

detenida por.....
julgada em 31/05/00
Orientador

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E COMPUTAÇÃO
DEPARTAMENTO DE MICROONDAS E ÓPTICA

**Refinamento Adaptativo de Malhas 2D e 3D para a Solução
Numérica de Problemas de Propagação Eletromagnética**

Autor:

Emir Baude

Orientador:

Hugo Enrique Hernández-Figueroa

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Banca Examinadora:

Prof. Dr. Hugo Enrique Hernández-Figueroa - DMO/FEEC/UNICAMP

Prof. Dr. Rui Fragassi Souza - DMO/FEEC/UNICAMP

Prof. Dr. Aldário Chrestani Bordonalli – DMO/FEEC/UNICAMP

Prof^a. Dr^a. Wu Shin-Ting - DCA/FEEC/UNICAMP

Prof. Dr. Luiz Lebensztajn - USP

Prof. Dr. Antônio Romeiro Sapienza - UERJ

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade
Estadual de Campinas como parte dos requisitos exigidos para a obtenção do grau de
DOUTOR EM ENGENHARIA ELÉTRICA

Campinas, Maio de 2000



418510009

UNICAMP
 N.º CHAMADA: 17 Unicamp
B324r
 V. 42747 Ex. 42747
 TOMBO BC/ 42747
 PROC. 16/278/00
 C ☐ D ☒
 PREÇO R\$ 11,00
 DATA 18/10/00
 N.º CPD

CM-00144276-5

FICHA CATALOGRÁFICA ELABORADA PELA
 BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

B324r Baude, Emir
 Refinamento adaptativo de malhas 2D e 3D para a
 solução numérica de problemas de propagação
 eletromagnética / Emir Baude.--Campinas, SP: [s.n.],
 2000.

Orientador: Hugo Enrique Hernández-Figueiroa.
 Tese (doutorado) - Universidade Estadual de
 Campinas, Faculdade de Engenharia Elétrica e de
 Computação.

1. Geração numérica de malhas. 2. Elementos finitos.
 3. Eletromagnetismo. I. Hernández-Figueiroa, Hugo
 Enrique. II. Universidade Estadual de Campinas.
 Faculdade de Engenharia Elétrica e de Computação. III.
 Título.

Resumo

Esta tese apresenta um novo algoritmo para o refinamento eficiente de malhas de elementos triangulares. Este tipo de malhas podem ser utilizadas por diversos métodos numéricos tais como o método dos elementos finitos nodais e de aresta, o método dos momentos, etc... Este novo algoritmo é baseado em uma combinação entre as técnicas de refinamento através da bissecção e a técnica de Delaunay. Apresenta-se também a extensão desta técnica para o refinamento de malhas tridimensionais superficiais e volumétricas. Diversas simulações de problemas eletromagnéticos foram conduzidas, as quais confirmam a eficácia de nosso algoritmo na geração de malhas com elementos totalmente conformes e compatíveis.

Abstract

A new algorithm for the efficient refinement of triangular element meshes is presented in this thesis. This kind of meshes are useful for several numerical methods like nodal and edge finite elements, method of moments, etc.. This new algorithm is based on a combination of bisection and the Delaunay refinement techniques. An extension of this technique to the refinement of volumetric and superficial tridimensional meshes is also presented. Several simulations of electromagnetic problems were performed, which confirm the effectiveness of the present algorithm in the generation of meshes with fully conformal and compatible elements.

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Agradecimentos

Gostaria de agradecer a meus pais e irmãos pelo apoio dado durante todos os momentos.

Agradeço também ao meu orientador, o Prof. Hugo, pelo apoio, ajuda e compreensão durante a construção desta tese.

Agradecimentos a todos alunos do DMO com os quais tive contato durante o tempo que durou este trabalho.

Agradeço também aos colegas Marcelo e Antônio, por desenvolverem programas que me ajudaram no cálculo numérico tridimensional.

Agradecimentos à Sueli por aguentar todos os momentos de mau humor que fatalmente ocorrem durante a realização de um trabalho como este.

Agradeço à FAPESP pelo apoio financeiro recebido durante este projeto.

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Sumário

1 Introdução	1
1.1 Simulação Computacional.....	2
1.2 Métodos de Análise Eletromagnética.....	3
1.3 Objetivo da Tese.....	5
2 Malhas Bidimensionais e Refinamento	9
2.1 Propriedades Desejadas em Malhas de Elementos Finitos.....	11
2.2 Funções de Densidade e Estimativas de Erro.....	14
2.3 Técnicas de Refinamento Adaptativo.....	19
2.3.1 Técnica de Refinamento Baricêntrica.....	20
2.3.2 Técnicas de Refinamento pela Bisseção de Elementos.....	23
2.3.3 Técnica Proposta.....	31
2.3.3.1 Desenvolvimento do Algoritmo.....	31
2.3.3.2 Implementação Final do Algoritmo.....	34
3 Malhas Tridimensionais Superficiais	39
3.1 Representação das Superfícies.....	40
3.2 Modelamento das Superfícies.....	42
3.3 Troca de Diagonais em Malhas Superficiais.....	45
3.4 Implementação do Algoritmo de Refinamento.....	46

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

4 Malhas Tridimensionais Volumétricas	51
4.1 Refinamento Tridimensional.....	53
4.2 Transformações Topológicas.....	56
4.3 Implementação do Algoritmo.....	63
 5 Métodos Numéricos	 69
5.1 – As Equações de Maxwell e a Equação de Onda.....	70
5.2 – Método dos Elementos Finitos.....	72
5.2.1 – Aplicações do Método dos Elementos Finitos.....	75
5.3 – Método dos Elementos de Aresta.....	77
5.4 – Método dos Momentos.....	81
 6 Resultados	 87
6.1 – Recursos Utilizados.....	88
6.2 – Resultados do Refinamento Bidimensional.....	89
6.2.1 – Estudo Comparativo.....	90
6.2.1.1 – Refinamento Baricêntrico.....	91
6.2.1.2 – Refinamento pela Bissecção da Maior Aresta – Nambiar.....	93
6.2.1.3 – Refinamento pela Bissecção da Maior Aresta – Rivara.....	96
6.2.1.4 – Refinamento pela Técnica Proposta.....	98
6.2.1.5 – Comparações entre os resultados.....	99
6.2.2 – Exemplos de Refinamento.....	102
6.2.2.1 – Refinamento de um guia tipo microfita.....	102
6.2.2.2 – Refinamento de um acoplador óptico.....	105
6.2.3 – Resultados Numéricos.....	109
6.2.3.1 – Análise de um guia de onda tipo Costela (‘Rib’).....	109
6.2.3.2 – Análise de um guia de onda ‘strip’ não-linear.....	114
6.3 – Resultados do Refinamento Tridimensional Superficial.....	120
6.3.1 – Exemplos de Refinamento.....	121
6.3.2 – Estudo Numérico.....	126

6.4 – Resultados do Refinamento Tridimensional Volumétrico.....	130
6.4.1 – Exemplos de Refinamento.....	130
6.4.2 – Exemplo Numérico.....	134
7 Conclusões	139
7.1 – Técnicas de Refinamento.....	139
7.2 – Sugestões para trabalhos futuros.....	140
A Estruturas de Dados e Arquivos de Entrada	143
A.1 – Estruturas de Dados.....	143
A.1.1 – Estruturas de Dados para o Refinamento Bidimensional.....	144
A.1.2 – Estruturas de Dados para o Refinamento 3D Superficial.....	145
A.1.3 – Estruturas de Dados para o Refinamento 3D Volumétrico.....	146
A.2 – Arquivos Iniciais.....	148
B Interface Gráfica	155
Referências	163

Lista de Figuras

Figuras

1.1 – Tipos de malhas de elementos finitos. (a) Malha com elementos triangulares e (b) malha com elementos retangulares	4
2.1 – Exemplo de malha com elementos não conformes	12
2.2 – Exemplos de elementos degenerados	13
2.3 – Transição abrupta entre elementos pequenos e grandes	14
2.4 – Fluxograma de um programa que utiliza as funções de densidade para monitorar o refinamento	15
2.5 – Fluxograma de um programa que utiliza estimativas de erro para monitorar o refinamento	19
2.6 – Subdivisão de elementos. (a) Baricêntrica e (b) em uma fronteira externa	21
2.7 – Bissecção da aresta 23 (linha tracejada), quando tratamos de interfaces curvas	21
2.8 – Exemplo de troca de diagonais	22
2.9 – Suavização, ou relaxação, da posição do nó P	23
2.10 – Bissecção através do algoritmo apresentado por Rivara, em [Rivara89]	24
2.11 – Divisão de um elemento em 4 elementos	25
2.12 – Exemplificação do processo de bissecção pelo maior lado	26
2.13 – Refinamento através da bissecção+LEPP. (a) Malha inicial, (b)-(f) passos do processo e (g) malha final	29
2.14 – Exemplo de refinamento com o segundo algoritmo de Rivara	30

2.15 – Exemplo de visita por adjacência, para se achar o elemento com a maior aresta em determinada região	33
2.16 – Fluxograma do algoritmo de refinamento	36
2.17 – Exemplo de visita por adjacência acelerada	37
3.1 – Decomposição de um modelo 3D em seus componentes geométricos	40
3.2 – Refinamento (a) conforme e (b) não-conforme em superfícies tridimensionais	41
3.3 – Mapeamento de um retângulo tridimensional no espaço paramétrico bidimensional e vice-versa	42
3.4 – Exemplo de troca de diagonais imprópria	45
3.5 – Troca de diagonais através da projeção	46
3.6 – Exemplo do processo necessário para a troca de diagonais	48
4.1 – Exemplos de elementos degenerados. (a) Cunhas, (b) cume e (c) lascas	53
4.2 – Um tetraedro degenerado do tipo lasca	54
4.3 – Formas de inserir um novo nó E no tetraedro formado pelos nós $ABCD$. As linhas pretas representam as arestas do tetraedro original e as linhas vermelhas as arestas resultantes da subdivisão	55
4.4 – Subdivisão de 4 tetraedros em 8 tetraedros, através da inserção de um novo nó G na aresta AF	55
4.5 – Elementos tridimensionais não conformes	56
4.6 – Exemplos de trocas de diagonais impróprias em tetraedros	57
4.7 – Troca de diagonais externa	58
4.8 – Troca de diagonais interna em um conjunto de 4 tetraedros	59
4.9 – Transformação topológica na qual fazemos a transformação de dois elementos em três e vice-versa	60
4.10 – Poliedro formado por cinco elementos ao redor da aresta AG e as cinco configurações possíveis para transformação	61
4.11 – Relaxação dos nós em três dimensões	62
4.12 – Classificação dos nós de acordo com sua posição na malha: 0 – internos, 1 – superficiais, 2 – de aresta e 3 – de canto	62
4.13 – Fluxograma da implementação do programa de refinamento 3D volumétrico	64
4.14 – Fluxograma da subrotina de refinamento	66

4.15 – Fluxograma do processo de relaxação	67
5.1 – Domínio unidimensional de comprimento L dividido em N elementos, e com N_p pontos nodais	73
5.2 – Funções de interpolação linear unidimensional	74
5.3 – Elemento triangular de aresta	78
5.4 – Funções de base vetoriais associadas às arestas de um elemento triangular: (a) N_1^e , (b) N_2^e e (c) N_3^e	80
5.5 – Figura ilustrando a superfície S , e alguns componentes do problema	83
5.6 – Onda plana incidente e seus componentes	84
6.1 – Malha inicial utilizada no estudo comparado de refinamento	90
6.2 – Refinamento utilizando a técnica baricêntrica.(a) Malha resultante e (b) detalhe da região do canal	92
6.3 – Gráficos das distribuições de (a) qualidade e (b) ângulos internos em função das quantidades normalizadas para a malha apresentada na Figura 6.2	93
6.4 – Refinamento utilizando a técnica de bissecção com lista ordenada. (a) Malha final e (b) detalhe do canal interno	94
6.5 – Distribuição da (a) qualidade e (b) dos ângulos internos da malha final, mostrada na Figura 6.4	95
6.6 – Refinamento utilizando a técnica de Bissecção+LEPP com troca de diagonais. (a) Malha final e (b) detalhe da região central	96
6.7 – Distribuição da (a) qualidade e (b) dos ângulos internos da malha final, mostrada na Figura 6.6	97
6.8 – Refinamento utilizando a técnica proposta. (a) Malha final e (b) detalhe	98
6.9 – Distribuição (a) da Qualidade e (b) dos ângulos internos da malha final	99
6.10 – Comparação entre os tempos de execução das técnicas apresentadas: 1 – Técnica baricêntrica, 2 – Algoritmo de bissecção + lista ordenada, 3 – Algoritmo de bissecção + LEPP +troca de diagonais e 4 – Técnica proposta	100
6.11 – Estrutura de um guia de ondas tipo microfita blindada	102
6.12 – (a) Malha inicial para o guia tipo microfita e (b) um detalhe de uma das regiões centro-laterais que contém os elementos de pior aspecto	103

6.13 – (a) Malha resultante do refinamento do guia tipo microfita, Figura 6.11 e (b) detalhe da região centro-lateral, mostrada na figura 6.12(b), após o refinamento	104
6.14 – Gráficos das distribuições (a) da qualidade em relação ao número normalizado de elementos e (b) dos ângulos internos em relação à quantidade normalizada de ângulos internos	105
6.15 – Estrutura de um acoplador a fibra óptica	106
6.16 – Malha inicial utilizada para o acoplador óptico	106
6.17 – Malha final para o refinamento adaptativo de um acoplador óptico	107
6.18 – Detalhe da região central da Figura 6.17	108
6.19 – Gráficos das distribuições (a) da qualidade em relação ao número normalizado de elementos e (b) dos ângulos internos em relação à quantidade normalizada de ângulos internos	108
6.20 – Guia de onda tipo costela e suas dimensões, em micrômetros	110
6.21 – (a) Malha inicial (116 elementos e 71 nós) e (b) malha da solução mais exata (4322 elementos e 2192 nós)	110
6.22 – (a) Malha final com 3462 elementos e 1744 nós. (b) Detalhe da região central. Nas Figuras 6.22(c) e (d) mostramos detalhes da região lateral central da malha inicial, Figura 6.21(a), e da malha final, Figura 6.22(a), respectivamente	111
6.23 – (a) Qualidade e (b) Distribuição de ângulos internos da malha mostrada na Figura 6.22(a)	112
6.24 – Gráfico da convergência do índice efetivo	113
6.25 – Campo elétrico tridimensional e curvas de contorno	113
6.26 – Estrutura do guia de onda não-linear	114
6.27 – (a) Malha inicial e (b) malha de controle	115
6.28 – Malha utilizada para (a) o cálculo do campo inicial e (b) detalhe da região central.....	116
6.29 – Malha utilizada para (a) o cálculo do campo final e (b) detalhe da região central.....	117
6.30 – Gráficos da distribuição de qualidade em relação ao número normalizado de elementos das malhas mostradas nas Figuras 6.28 e 6.29, respectivamente	118

6.31 – Gráficos da variação do índice efetivo em função da potência. (a) Resultado obtido por nós e (b) por Li	118
6.32 – Distribuições de campo obtidas utilizando as malhas mostradas nas Figuras 6.28 e 6.29, respectivamente	119
6.33 – Curvas equipotenciais para os campos (a) inicial e (b) final	120
6.34 – Refinamento superficial de uma estrutura quadrada. (a) Malha inicial e (b) malha final	121
6.35 – Distribuição (a) da qualidade e (b) dos ângulos internos da malha mostrada na Figura 6.34(b)	122
6.36 – Refinamento de uma calota esférica. (a) Malha inicial, (b) Visão lateral da malha final e (c) Visão superior da malha final	123
6.37 – Distribuição (a) da qualidade e (b) dos ângulos internos da malha final	124
6.38 – Refinamento de um cilindro, (a) Malha inicial e (b) Malha final	124
6.39 – Distribuição (a) da qualidade e (b) dos ângulos internos para a malha mostrada na Figura 6.38(b)	125
6.40 – Malha final para o cálculo numérico	126
6.41 – Distribuição (a) da qualidade e (b) dos ângulos internos da malha mostrada na Figura 6.40	127
6.42 – Vista lateral da distribuição superficial normalizada de corrente na esfera	128
6.43 – Distribuição superficial normalizada de corrente na esfera: (a) Vista Superior e (b) Vista inferior	128
6.44 – Comparação entre os resultados para o cálculo da seção reta do espalhamento bi-estático de uma onda incidente E , Figuras (a) e (b), e H , Figuras (c) e (d). Os dados correspondentes às Figuras (a) e (c) foram obtidos por nós, enquanto que os dados das Figura (b) e (d), foram obtidos por Wang	129
6.45 – Refinamento de uma estrutura tetraédrica. (a) Malha inicial e (b) malha final	131
6.46 – Distribuições (a) da qualidade da malha em relação ao número de elementos e (b) do ângulo interno em relação à quantidade normalizada de ângulos internos	131
6.47 –(a) Malha inicial e (b) malha final para o estudo do refinamento de um cubo	132
6.48 – Distribuição (a) da qualidade e (b) dos ângulos internos da malha final	133
6.49 – Tempo de processamento versus Quantidade de elementos	133

6.50 – Cavidade retangular e dimensões	134
6.51 – Refinamento de uma cavidade retangular tridimensional. (a) Malha Inicial e (b) Malha Final	135
6.52 – Distribuição (a) da qualidade e (b) dos ângulos internos na malha final	136
6.53 – Convergência dos resultados em relação ao resultado analítico, em números normalizados	136
A.1 – Representação do elemento E em nossa estrutura de dados	144
A.2 – Representação do elemento E_1 e da aresta A_2	147
B.1 – Interface gráfica e exemplo de malha	156
B.2 – Menu <i>File</i> e seus componentes	157
B.3 – Menu <i>Edit</i> e seus componentes	157
B.4 – Diálogo de edição de preferências	158
B.5 – Diálogo para a edição das gaussianas	159
B.6 – Menu <i>Window</i> e seus componentes	160
B.7 – Menu <i>Mesh</i> e seus componentes	160
B.8 – Menu <i>Help</i> e seus componentes	161
B.9 – Diálogo de Rotação da estrutura	161
B.10 – Diálogo para visualização de um determinado elemento, e um exemplo da apresentação do elemento	162

Tabelas

6.1- Convergência dos resultados em relação ao resultado analítico, em números normalizados	137
---	-----

Capítulo 1

Introdução

A simulação computacional é uma ferramenta poderosa para o estudo e projeto de equipamentos e seus componentes pois permite ao projetista conhecimento a respeito das características do objeto de estudo sem a necessidade de gasto de material e de tempo comuns à construção de um protótipo, tendo ainda a possibilidade de se obter certos parâmetros difíceis de serem medidos em outras condições. Cada vez mais a prévia modelagem, usando técnicas numéricas apropriadas para a simulação em computadores, é mais efetiva que a repetida construção de protótipos e seus testes em laboratório, embora a construção de um protótipo final seja um elemento necessário para a confirmação da exatidão das simulações numéricas e da viabilidade de produção da estrutura.

Da mesma forma que estas simulações são utilizadas na mecânica, aerodinâmica e dinâmica de fluidos, seu uso na engenharia elétrica é feito com o objetivo de auxiliar no

desenvolvimento de uma grande gama de projetos, desde a pesquisa de máquinas elétricas até fibras óticas e componentes de ótica integrada.

A modelagem numérica de estruturas eletromagnéticas envolve, tipicamente, a solução das Equações de Maxwell nas formas integral ou diferencial em um espaço finito. Exceto para os casos mais simples, isto significa a quebra do domínio em subdomínios menores que serão descritos por um conjunto de equações ou incógnitas mais simples. Para se chegar à solução do problema como um todo, estas equações são resolvidas juntas, providenciando a solução final desejada. O número e complexidade das equações a serem resolvidas dependem do tipo de discretização utilizada e da complexidade do problema, isto é, se o objeto não é homogêneo, ou se possui não-linearidades. Tipicamente, quanto mais complexo o problema e/ou o método escolhido para sua resolução, maior será o esforço computacional necessário para resolvê-lo.

O melhor método a ser utilizado na simulação do comportamento de um sistema físico é uma função das características do conjunto de equações que governam o modelo e dos recursos computacionais disponíveis (incluindo 'software', 'hardware', etc...) [Rieder86].

1.1 - Simulação Computacional

O uso da modelagem computacional para avaliar os parâmetros de determinado componente pode proporcionar um grau de exatidão variável, dependendo de alguns fatores. Primeiramente é preciso descrever convenientemente o problema. Um modelo com pouca exatidão em algum ponto pode gerar uma solução que corresponda a outro problema diferente daquele que queremos estudar. Depois devemos escolher o método de resolução que melhor se adapte ao problema. Alguns métodos fazem simplificações do problema, de forma que ele seja mais facilmente resolvido pelo computador, o que pode incorrer em

limitações na solução final. Finalmente, devemos analisar as facilidades computacionais disponíveis. Quanto maior o poder computacional disponível maior será a exatidão da solução, embora com a crescente evolução dos computadores pessoais, é possível utilizá-los para se resolver uma grande quantidade de problemas sem uma grande demanda de tempo [Dean96].

Um dos principais interesses atualmente é não mais elaborar programas que simplesmente efetuem o cálculo numérico de maneira isolada, mas desenvolver sistemas completos que possibilitem a utilização simples e racional do método escolhido por qualquer engenheiro projetista. Tais sistemas são compostos basicamente de três blocos:

- a) Pré-processamento: Nesta etapa, devemos fornecer a estrutura a ser analisada, bem como as características do material de que são compostas.
- b) Processamento: É o cálculo numérico propriamente dito, utilizando-se os dados gerados na primeira etapa do processo.
- c) Pós-processamento: Nesta etapa apresentamos os resultados da forma o mais eficiente possível, de forma a facilitar a sua análise de forma correta pelo projetista.

A etapa **b** do processo certamente é a parte mais importante de todo o processo pois nela é feita o modelamento físico - numérico do problema. No entanto, a inexistência das etapas de pré e pós-processamento dificultam o trabalho do usuário pois, por exemplo, a elaboração manual de uma malha a ser analisada através do Método de Elementos Finitos, na etapa de processamento, é uma tarefa extremamente complexa e sujeita a erros [Bastos92].

1.2 - Métodos de Análise Eletromagnética

Entre os métodos existentes para se fazer o processamento podemos citar três que são bastante utilizados: Método dos Elementos Finitos Nodais, Método dos Elementos Finitos de Aresta e Método dos Momentos. No Capítulo 5 procuraremos apresentar uma breve introdução a cada um deles. Entretanto, o Método de Elementos Finitos é reconhecido atualmente como a melhor técnica numérica [Jin93, Koshiba92], e por isso nos referenciaremos principalmente a ele neste trabalho.

Um fator comum entre eles, é que em todos eles são feitas pequenas subdivisões do domínio original, que podem ter tamanho variável dentro do domínio. Ao conjunto total de sub-regiões nós damos o nome de malha de elementos finitos, conforme exemplos das Figuras 1.1. Estas malhas podem, por exemplo, ser compostas de elementos triangulares ou retangulares, conforme Figuras 1.1(a) e 1.1(b). Detalhando os componentes da malha apresentada na Figura 1.1(a), cada triângulo corresponde a um elemento e o ponto de encontro de duas arestas deste triângulo é denominado nó, ou ponto nodal.

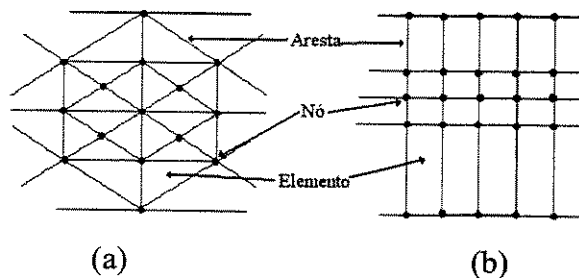


Figura 1.1 – Tipos de malhas de elementos finitos. (a) Malha com elementos triangulares e (b) malha com elementos retangulares.

As malhas triangulares são as mais comuns devido à sua melhor adaptação a domínios de forma complexa, enquanto que as malhas retangulares são mais utilizadas em simulações através do método de diferenças finitas.

A geração de malhas geometricamente versáteis e computacionalmente eficientes é responsável pela otimização dos recursos computacionais no uso do Método dos Elementos Finitos. Considerando que o ambiente computacional, em constante evolução, tem um papel crucial nas técnicas de geração de malhas, não resta dúvidas que o estudo das mesmas é um assunto moderno e dinâmico.

Recentemente, novos conceitos em computação gráfica impulsionaram o surgimento de novas estratégias na geração de malhas de elementos finitos. Entretanto na área que é de nosso interesse, eletromagnetismo de altas frequências, pouco tem sido feito, o que torna necessários estudos nessa direção adaptando técnicas recentemente desenvolvidas em outras áreas e propondo novas estratégias originadas a partir de nossos problemas [Yagel95].

A adaptividade destas malhas também é outro fator de extremo interesse, pois nos permite concentrar mais elementos em regiões de maior interesse, deixando as outras regiões com a quantidade mínima suficiente para o correto funcionamento do programa de resolução numérica. Com isto, um número menor de incógnitas é necessário para descrever o problema, implicando em um menor tempo de processamento.

Até o momento tratamos apenas do caso bidimensional, mas a mesma análise pode ser feita para o caso tridimensional, apenas substituindo-se os elementos triangulares por tetraedros e os elementos retangulares por hexaedros.

1.3 - Objetivo da Tese

O ponto central desta tese é desenvolver algoritmos para geração de malhas adaptativas de elementos finitos, para uso em estudos de propagação de ondas eletromagnéticas em domínios bidimensionais e tridimensionais. Tais algoritmos devem ser

robustos e altamente eficientes, tanto nos critérios de qualidade geométrica e suavidade da malha quanto no tempo de computação. Estes programas devem ainda ter a capacidade de se adaptarem a estruturas complexas.

O algoritmo em questão é baseado em dois algoritmos bastante conhecidos: a técnica de partição em 2 triângulos, ou bissecção, pela aresta mais longa, e a técnica da partição em 3 triângulos, ou baricêntrica. O programa preserva as qualidades de ambas as técnicas e contorna suas deficiências.

Infelizmente, durante o processo de validação do algoritmo desenvolvido para contornar um defeito da técnica de bissecção, encontramos um algoritmo semelhante, que foi desenvolvido por Rivara [Rivara97a,b], aproximadamente na mesma época que o nosso e levando em conta as mesmas premissas, o que levou a resultados semelhantes. O que não é de todo mau, pois indica que o caminho traçado por nós era acertado. Maiores detalhes desta semelhança entre ambas as técnicas são descritos na Seção 2.3.2. Felizmente, algumas modificações em nosso algoritmo se revelaram superiores aos outros, produzindo bons resultados.

No Capítulo 2 nós falaremos a respeito das malhas bidimensionais, as características que as malhas de elementos finitos triangulares devem ter para produzir bons resultados, técnicas de refinamento adaptativo (formas de partição, transformações geométricas, critérios de erro, etc.) e, por último, apresentaremos o algoritmo de refinamento por nós desenvolvido que apresenta vantagens em relação aos outros algoritmos estudados.

No Capítulo 3 veremos a parte de refinamento tridimensional superficial, ainda usando elementos triangulares, e no Capítulo 4 a parte de refinamento tridimensional volumétrico, usando tetraedros. Em ambos os casos, os algoritmos foram desenvolvidos a partir do algoritmo desenvolvido para o caso bidimensional. Falaremos, ainda, sobre as transformações que se tornam necessárias nos dois casos.

No Capítulo 5 apresentaremos alguns comentários sobre os métodos utilizados nos cálculos (elementos finitos, elementos de contorno, momentos) usando as malhas bidimensionais e tridimensionais.

No Capítulo 6 mostraremos os resultados obtidos para os três tipos de refinamento (2D, 3D superficial e 3D volumétrico), avaliando tanto o aspecto geométrico das malhas, quanto a precisão das simulações numéricas conduzidas utilizando malhas obtidas com o auxílio de nosso programa. Finalmente no Capítulo 7 chegamos às conclusões finais e sugestões para trabalhos futuros. Nos Apêndices A e B veremos, ainda, detalhes sobre a estrutura de dados utilizada, formato dos arquivos de entrada e sobre a interface gráfica desenvolvida.

Capítulo 2

Malhas Bidimensionais e Refinamento

O Método dos Elementos Finitos (MEF) começou a ser desenvolvido na década de 1950 com o objetivo de solucionar problemas ligados à Engenharia Mecânica, mas logo era proposta a sua utilização no estudo de campos eletromagnéticos [Bastos92], principalmente nos problemas associados ao regime de baixas frequências, enquanto que os casos relativos a altas frequências passaram a ter um maior enfoque a partir da década de 80. Esta técnica numérica parte do princípio de que o domínio pode ser decomposto em pequenas regiões, e uma formulação adequada é aplicada a cada região ou elemento (ver Capítulo 5). Assim, o conjunto de soluções relativo a estes elementos fornece a solução global do problema. É evidente que quanto menores forem estes elementos, mais a solução discretizada se aproxima da solução ideal. Naturalmente, em muitos problemas o grau de discretização necessário para obter um certo nível de acuracidade varia sobre o domínio do problema. É aqui que aparece uma das grandes virtudes do Método dos Elementos Finitos: a

possibilidade de podermos usar malhas com densidade variável. Além disto o método propicia o refinamento da malha através de algoritmos adaptativos [Fernandez93].

A técnica de refinamento adaptativo de malhas de elementos finitos consiste em primeiramente obter-se uma malha inicial com poucos elementos e que descreva adequadamente a estrutura a ser analisada. Esta malha inicial pode ser gerada manualmente ou através de uma grande variedade de métodos automáticos, entre os quais podemos destacar o método da colocação de nós, decomposição recursiva, frente progressiva ('advancing front'), baseado em grades, etc... . Como o objetivo desta tese é o estudo de algoritmos de refinamento adaptativo de malhas de elementos finitos, não trataremos deles, remetendo o leitor interessado às nossas referências [Borouchaki97, Fuchs97, Ho-Le88, Lee99, Shephard88].

A partir da malha inicial define-se uma função apropriada para o domínio que é, então, monitorada elemento a elemento. Aqueles elementos que não satisfazem certa condição limite, definida pelo usuário, são então subdivididos utilizando-se alguma das técnicas descritas na Seção 2.3. O procedimento é feito recursivamente até que se atinja o número de elementos desejado, ou até que todos os elementos satisfaçam a um critério de parada, que pode variar conforme a técnica utilizada (Seção 2.2). A característica mais marcante do refinamento é que novos nós são adicionados à malha e nenhum é removido.

As funções de monitoramento descritas no parágrafo anterior podem ser de dois tipos: *a priori* e *a posteriori*. Na primeira utilizamos funções de densidade pré-definidas, pelo usuário para gerenciar o refinamento [Fernandez93, Figueroa96]. Assim, partindo de uma malha inicial com poucos elementos, é feito o refinamento da malha inicial de forma controlada pelas funções de densidade pré-definidas, até se atingir o número de elementos desejado.

No outro caso, usamos estimativas de erro [Fuenmayor96, Ida94, Itoh96]. O método consiste, basicamente, no uso de uma certa malha inicial para fazermos o cálculo desejado. Após este passo inicial, calculamos o erro (local ou global) da solução encontrada com esta

malha e utilizamos estes dados para monitorar o refinamento de uma nova malha, a partir da malha inicial, se isto se fizer necessário. O processo é feito recursivamente até que se obtenha o grau de exatidão necessário.

A desvantagem do primeiro caso é que a exatidão do resultado obtido depende fortemente da prática do próprio usuário e do seu conhecimento intrínseco do problema, necessários para se escolher as funções apropriadas a fim de modelar o problema de forma conveniente. Quanto ao segundo caso, não é necessária a experiência prévia do projetista para selecionar as áreas do domínio a serem refinadas, porém existe a necessidade de uma malha de elementos finitos mais densamente refinada para ser usada como controle de erro. Na Seção 2.2 exporemos ambas as técnicas com maior detalhamento.

Um fator importante no refinamento adaptativo certamente é o tempo de computação; porém, outros fatores também são igualmente importantes para que a malha tenha certas propriedades que tornem os resultados obtidos mais exatos e dignos de confiança. A verificação destas propriedades se torna mais difícil à medida que aumentamos o grau de complexidade da estrutura. Mostraremos algumas destas propriedades na Seção a seguir.

2.1 – Propriedades Desejadas em Malhas de Elementos Finitos

Infelizmente o refinamento adaptativo é um problema mais difícil do que pode parecer à primeira vista. A necessidade de se obter uma discretização coerente, gerando uma malha computacionalmente eficaz, impõe certos fatores de qualidade à malha, que precisam ser satisfeitos.

Um destes fatores é a conformidade da malha. Uma dada malha é dita conforme quando não ocorrem casos em que um ponto interno ao domínio divide uma aresta de um

dado elemento, mas não divide o elemento adjacente a esta aresta. Portanto, para que a malha seja compatível, é preciso que cada aresta possua no mínimo 1 e no máximo 2 elementos associados a ela, e que cada elemento seja definido por apenas 3 arestas, no caso de triângulos. Um exemplo de malha não conforme é mostrado na Figura 2.1, onde vemos que as arestas AC e CB são apenas parte da aresta AB . Nela vemos que a aresta AB está associada a 3 elementos, e que o elemento $EL.1$ pode ser descrito por 4 arestas.

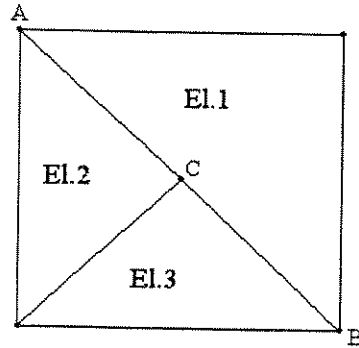


Figura 2.1 – Exemplo de malha com elementos não conformes.

Existem técnicas derivadas do método de elementos finitos que permitem esta não conformidade da malha, porém estes elementos não são muito utilizados, pois eles podem por em risco a convergência da solução numérica do problema. Embora tais métodos não-conformes possam gerar malhas de forma mais fácil e com uma aparência mais regular, o problema do erro numérico persiste e desencoraja o uso de tais técnicas[Hitschfield95].

Outro fator de qualidade que a malha deve ter, é a qualidade individual de seus elementos, α . Esta qualidade pode ser medida por uma Equação muito simples, dada por

$$\alpha = 4\sqrt{3} \frac{A}{\sum_{i=1}^3 d_i^2} \quad (2.1)$$

onde A é a área do elemento, e d_i é o comprimento de cada aresta deste elemento. O valor de qualidade máximo é igual a 1,0, no caso de um triângulo equilátero, e decai para valores

próximos a zero para os casos de triângulos degenerados, ou seja, triângulos em que ao menos um dos ângulos internos está próximo de zero. Na Figura 2.2 vemos dois exemplos de elementos degenerados.

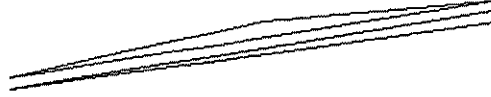


Figura 2.2 – Exemplos de elementos degenerados.

Estes elementos degenerados devem ser evitados, pois a ocorrência de ângulos muito agudos ou muito obtusos além de degradar a qualidade geométrica da malha, contribuem para diminuir a qualidade numérica da solução obtida através da simulação. Isto pode ser verificado se lembrarmos que os erros de aproximação na variável de estado u , E_0 , e na derivada primeira de u , E_1 , em um elemento finito de primeira ordem (ver seção 5.2) são dados pelas equações

$$E_0 = C_0 L^2 \text{Max} \left(\frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial x \partial y}, \frac{\partial^2 u}{\partial y^2} \right)$$

$$E_1 = C_1 \frac{L}{\sin \theta} \text{Max} \left(\frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial x \partial y}, \frac{\partial^2 u}{\partial y^2} \right)$$

nas quais L é o comprimento da maior aresta, θ é o maior ângulo e C_0 e C_1 são constantes. Através destas equações é possível ver que quanto maior o ângulo ou o comprimento das arestas, maior serão os erros (Oden72).

Diversos autores impõem um limite de 30 graus para o ângulo interno mínimo, embora a condição ideal ocorra com ângulos iguais a 60 graus [Shewchuk97, Rivara97]. Existem outros meios para se medir a ocorrência de triângulos degenerados, por exemplo a razão entre a maior e a menor aresta de um elemento, mas estes métodos conduzem aos mesmos resultados que obtemos com a medida da qualidade dos triângulos.

Outra característica desejada da malha é que a transição entre elementos pequenos e grandes ocorra de forma suave, caso contrário ocorre a formação de ‘frentes’ de elementos na malha, isto é, os elementos tendem a se agrupar em conjuntos de triângulos com aspecto geométrico semelhante, com uma transição abrupta entre eles. Na Figura 2.3 vemos um exemplo de transição abrupta. Esta transição abrupta pode levar a erros na solução numérica do problema.

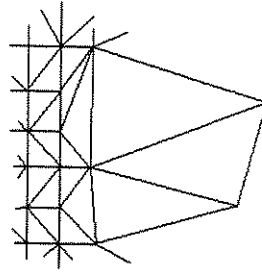


Figura 2.3 – Transição abrupta entre elementos pequenos e grandes.

Todos estes fatores devem ser levados em consideração para se obter uma malha compatível que não provoque erros numéricos na simulação. Técnicas de refinamento adaptativo foram desenvolvidas a fim de que estes fatores fossem respeitados e, ainda assim, produzissem malhas com bom aspecto geométrico.

2.2 – Funções de Densidade e Estimativas de Erro

Como dissemos no início deste Capítulo, para se fazer o refinamento adaptativo de qualquer malha é necessário que se definam funções apropriadas que monitorem este refinamento. Destacamos dois métodos que produzem resultados satisfatórios: as Funções de Densidade e as Estimativas de Erro.

A primeira delas é o uso de funções de densidade pré-definidas pelo usuário. Estas funções podem ser de qualquer tipo que o usuário escolher, embora o uso de gaussianas, na forma da Equação 2.2, seja uma boa escolha.

$$f(x, y) = Amp \cdot e^{-\left(\left(\frac{X-X_0}{Dx}\right)^2 + \left(\frac{Y-Y_0}{Dy}\right)^2\right)} \quad (2.2)$$

Nesta Equação Amp é a amplitude da gaussiana, X_0 , Y_0 são as coordenadas do centro desta gaussiana e Dx e Dy são as distâncias em relação ao centro nas quais a amplitude da gaussiana cai pela metade. Assim, se estivermos trabalhando com elementos de primeira ordem (ver Seção 5.2), o peso, ou densidade, individual w_i de cada elemento será dado por:

$$w_i = \frac{A}{3}(f_1 + f_2 + f_3) \quad (2.3)$$

onde A é a área do triângulo e f_i é o valor da função de densidade no nó i . A densidade total W da malha em questão é calculada através da integral da função de densidade escolhida sobre o domínio do problema, ou seja

$$W = \sum_{i=1}^M w_i = \sum_{i=1}^M \int f(x, y) dx dy \quad (2.4)$$

onde M é o número total corrente de elementos. Inicialmente, M corresponde ao número de elementos da malha primária, mas à medida que a malha é refinada, este número aumenta e, conseqüentemente, a densidade total se altera.

Como critério de parada, podemos especificar o número total de elementos desejado. Assim, se desejamos obter um número final de N elementos, todos com pesos individuais iguais ou menores que a condição limite I , então, de acordo com a Equação 2.4, a densidade total será $W=N \cdot I$. Utilizando isto, podemos dizer que o refinamento deve ser feito até que a Equação 2.5 seja satisfeita para todos os i elementos [Fernandez93].

$$w_i \leq I = \frac{W}{N} \quad (2.5)$$

Este método funciona muito bem, porém é necessário um certo conhecimento prévio do problema para escolher as funções de densidade que melhor se adaptem ao problema, além de posicionar corretamente essas funções, de forma que o domínio do problema realmente seja refinado onde é mais necessário. Na Figura 2.4 mostramos o fluxograma da técnica de refinamento utilizando as funções de densidade.

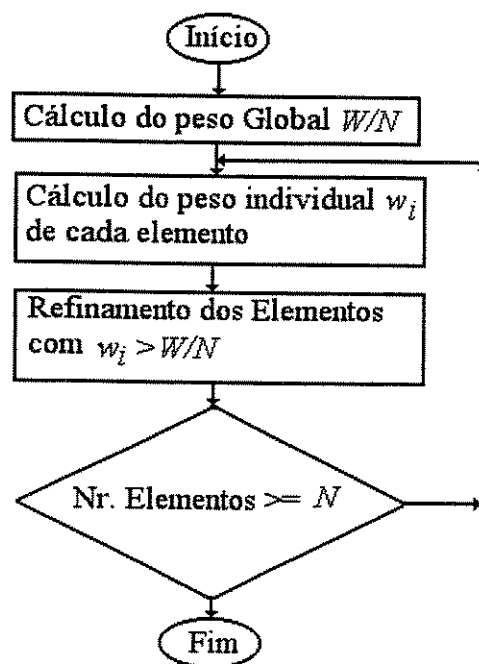


Figura 2.4 – Fluxograma de um programa que utiliza as funções de densidade para monitorar o refinamento.

Outra forma de se monitorar o refinamento da malha é o uso de estimativas de erro. Utilizamos as estimativas de erro a fim de controlarmos o grau de acerto de nossa simulação em relação a um cálculo exato. Supondo que temos um problema de valores de contorno linear da forma

$$Lu = f \quad \text{em} \quad \Omega \quad (2.6)$$

onde L é um operador, u é a escalar desconhecida, f é uma função dada e Ω é o domínio do problema. A solução é computada usando-se os dois conjuntos de soluções obtidos: Φ_{ex} e

Φ , ou seja, a solução exata e a solução que desejamos avaliar. A solução ótima é obtida quando a norma da solução Φ do sistema converge para a norma da solução Φ_{ex} [Ida94]. Podemos dizer então que a norma do erro de um elemento i qualquer é dado por:

$$\|e\|_i = \left(\int_{\Omega} e_i^* e_i \, d\Omega \right)^{1/2} \quad (2.7)$$

onde $e_i = (\Phi_{\text{ex}} - \Phi)_i$. O erro global na norma da energia é dado pela soma de todos os erros individuais da malha, ou seja:

$$\|e\|^2 = \sum_{i=1}^M (\|e\|^2)_i \quad (2.8)$$

aonde M é a quantidade corrente de elementos na malha.

Portanto, à medida que a malha é refinada o seu erro global relativo se altera, da mesma forma que ocorria no método anterior. A percentagem de erro global relativo, em relação à solução exata Φ_{ex} , é dada por

$$\eta = \left\{ \|e\| / \|\Phi_{\text{ex}}\| \right\} \cdot 100\% \quad (2.9)$$

Como, na prática, em muitos problemas a solução exata é impossível, ou impraticável de se obter, utiliza-se geralmente uma solução de maior exatidão no lugar da solução exata Φ_{ex} [Zhu91]. Habitualmente, esta solução é obtida a partir de uma malha com um número de elementos maior, ou com elementos de maior ordem, do que a malha que desejamos avaliar. Esta malha não precisa necessariamente estar refinada de forma mais densa em torno das regiões críticas, bastando gerar uma malha com um refinamento uniforme, porém com um bom número de elementos. Nestes casos, o nosso erro global relativo fica da forma

$$\eta = \left\{ \frac{\|e\|}{\left(\|\Phi\|^2 + \|e\|^2 \right)^{1/2}} \right\} \cdot 100\% \quad (2.12)$$

O erro admissível, em um dado elemento, é calculado utilizando-se o erro global relativo e o número de elementos

$$\|e\|_a = \frac{\eta}{100} \left\{ \frac{\|\Phi\|^2 + \|e\|^2}{M} \right\}^{1/2} \quad (2.10)$$

onde $\|\Phi\|^2 = \sum_{i=1}^M \|\Phi\|_i^2$

Da mesma forma que no método anterior, devemos definir um critério para o refinamento de cada elemento, e para isto podemos usar a taxa de variação entre o erro admissível, $\|e\|_a$, e o erro local, $\|e\|_i$.

$$\xi_i = \|e\|_i / \|e\|_a \quad (2.11)$$

Assim, se $\xi_i > 1$, o elemento i deve ser refinado para obtermos um novo valor que esteja dentro do limite estabelecido. Este fator serve para identificarmos as regiões do domínio do problema onde o refinamento é mais crítico e necessário, enquanto que o erro global relativo, η , age como um critério para a suspensão do refinamento. Muitas vezes, após uma série de iterações, o erro admissível atinge valores extremamente pequenos em relação ao erro admissível da primeira iteração, e podemos considerar estes erros como desprezíveis e suspender o refinamento.

Este método não necessita de um conhecimento prévio a respeito do problema, seu único defeito consiste na necessidade de elaboração de uma solução mais acurada para usarmos como comparação. Assim, um programa que utilize este método para o refinamento adaptativo, terá um fluxograma da forma mostrada na Figura 2.5. Nessa figura,

o fator $X\%$ representa o critério adotado para a suspensão do refinamento da malha, e Ea corresponde ao erro admissível, calculado com a ajuda da Equação 2.10.

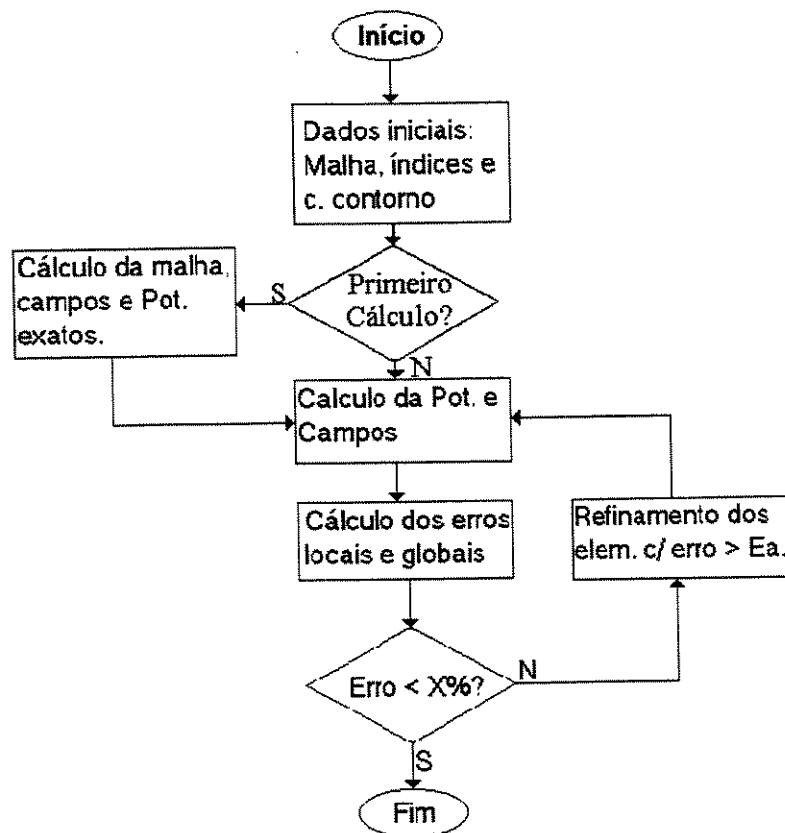


Figura 2.5 – Fluxograma de um programa que utiliza o método das estimativas de erro para monitorar o refinamento.

2.3 – Técnicas de Refinamento Adaptativo

Existem duas técnicas básicas para se fazer a divisão, ou partição, de um elemento triangular. A primeira divide, ou quebra, o elemento original em três novos elementos, e podemos chamá-la de técnica baricêntrica, pois adiciona o novo nó no centro do triângulo

[Fernandez93]. A segunda consiste na divisão do triângulo em dois ou quatro novos elementos, através da bissecção da(s) aresta(s) do elemento [Rivara89]. Algumas modificações foram introduzidas neste ultimo método, com o intuito de deixar o algoritmo mais robusto [Nambiar93, Rivara97a,b]. A seguir, veremos com detalhes cada uma destas técnicas, e ao final apresentaremos uma nova técnica, que conjuga as vantagens de cada uma delas.

2.3.1 – Técnica de Refinamento Baricêntrica

Esta técnica de refinamento começa pela criação de uma lista dos elementos a serem subdivididos. Esta lista pode ser obtida utilizando-se as funções de densidade ou as estimativas de erro, vistas na Seção 2.2. Assim, se dado triângulo estiver acima da densidade limite ele é adicionado à lista para posterior subdivisão, que pode ser feita de duas formas distintas. Quando se tratar de elementos que não possuam a maior aresta em uma fronteira (interna ou externa), a partição é feita através da inserção de um novo nó no baricentro do triângulo, usando-se as funções de densidade *a priori*, citadas na Seção 2.2, e conectando os nós já existentes a este novo nó, gerando três novos triângulos. Quando o elemento a ser dividido possuir a maior aresta em uma fronteira, o novo nó é inserido na aresta que pertence à fronteira, também utilizando-se das funções de densidade. Na Figura 2.6 vemos ambas as divisões descritas acima, onde o elemento original está em linha cheia e a subdivisão está em tracejado. Nos casos em que o elemento maior está posicionado em uma interface interna, é necessário que se divida o elemento que compartilha esta aresta com ele, para evitar problemas de conformidade, como vimos na Seção 2.1.

Como o processo de refinamento adaptativo é recursivo, nenhum elemento é dividido mais que uma vez a cada passo, para não deteriorar a qualidade da malha. Entretanto, é feita uma checagem nos elementos pertencentes às fronteiras para verificar se não ocorrem elementos de formato anômalo, isto é, que possuem ângulos muito agudos na fronteira. Caso ocorram incidências destes elementos, é feita a sua subdivisão através da

inserção de um nó na maior aresta, mesmo que este elemento esteja abaixo da condição limite imposta inicialmente.

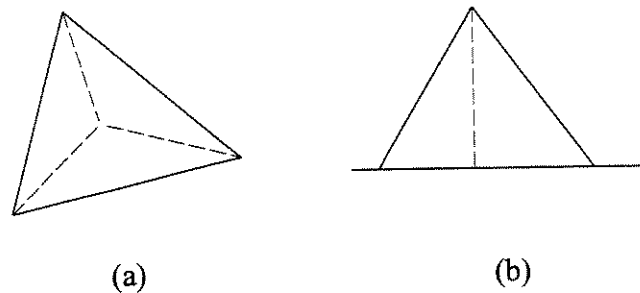


Figura 2.6 – Subdivisão de elementos. (a) Baricêntrica e (b) em uma fronteira externa.

Nos casos em que temos fronteiras curvas, internas ou externas, o programa deverá efetuar a divisão de forma tal que o novo ponto P fique eqüidistante dos nós da maior aresta e, ainda assim, verifique a Equação da curva, de acordo com os dados fornecidos ao programa. Em um exemplo simples, conforme Figura 2.7, se estivéssemos trabalhando com coordenadas polares, o ponto médio de um segmento de círculo seria calculado achando-se o ângulo θ correspondente ao ângulo formado entre as arestas 12 e 13 , e uma vez que o raio r do círculo é constante, basta acharmos o ângulo médio $\theta_m = \theta/2$. Assim, quando trabalhamos com o sistema de coordenadas cartesianas, basta uma simples transformação para o sistema polar, e de novo para o sistema cartesiano, para calcularmos a posição do novo nó. Existem diversos outros métodos para se fazer esta divisão, tanto para casos simples quanto para casos mais complexos, tais como elipses ou equações polinomiais. Nestes casos, a troca de diagonais não é aplicada às arestas pertencentes à(s) interface(s) pois isto poderia causar a destruição da mesma.

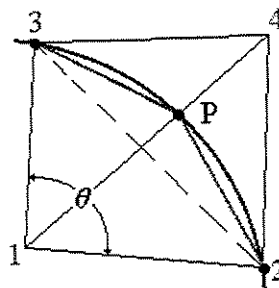


Figura 2.7 – Bisseção da aresta 23 (linha tracejada), quando tratamos de interfaces curvas.

Após terminada a lista e efetuadas todas as divisões, é utilizado o critério de troca de diagonais de Delaunay, também chamado de teste da circunferência, para melhorar o aspecto da malha. O procedimento para se fazer esta troca é bastante simples e pode ser acompanhado com a ajuda da Figura 2.8. Após acharmos o círculo circunscrito a um triângulo ABC , é feita uma verificação com o triângulo que compartilha a sua maior aresta, no caso o lado BC . Se o nó oposto ao nó A estiver situado dentro do círculo circunscrito, então a troca é recomendada. Já foi provado que esta troca maximiza o ângulo mínimo entre os seis ângulos formados pelos dois triângulos [Lee95]. Quando a aresta a ser trocada estiver em uma fronteira interna, a troca não é feita, pois destruiria esta interface. Ao invés disto, pode ser feita uma nova subdivisão, na forma descrita anteriormente. Este teste é aplicado a todos os elementos da malha.

Finalmente, verificamos se já atingimos o número desejado de elementos, caso contrário retomamos todo o processo, com a criação de uma nova lista.

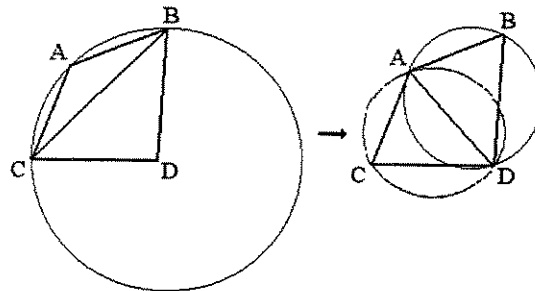


Figura 2.8 – Exemplo de troca de diagonais.

Este algoritmo produz malhas totalmente conformes e com elementos de bom aspecto geométrico. Devido à troca de diagonais, quaisquer elementos da malha inicial que tenham um aspecto ‘pobre’ podem ser melhorados após sucessivas divisões. Entretanto, uma das principais falhas deste algoritmo, é a produção de ‘frentes’ de elementos na malha, na forma descrita na Seção 2.1. Para se sanar este problema é possível fazer uma redução na distorção dos elementos, também conhecido como suavização ou otimização, que consiste na mudança de posição dos nós. O método Laplaciano é o método mais utilizado, embora novas aproximações tenham sido testadas com bons resultados [Canann93,

Freitag95], e consiste na determinação da nova posição do nó através da média das coordenadas dos nós conectados a ele.

$$P^* = (1/n) \sum_{i=1,2,\dots,n} P_i \quad (2.13)$$

onde P^* é a nova coordenada do nó sendo considerado, e P_i são as posições dos n nós conectados a P^* . Na Figura 2.9 vemos um exemplo da utilização do algoritmo. Na parte (a) vemos um nó P e suas conexões com os seus nós vizinhos, enquanto que na Figura 2.9(b), vemos o ponto P^* nas coordenadas resultantes da média laplaciana das coordenadas dos nós vizinhos.

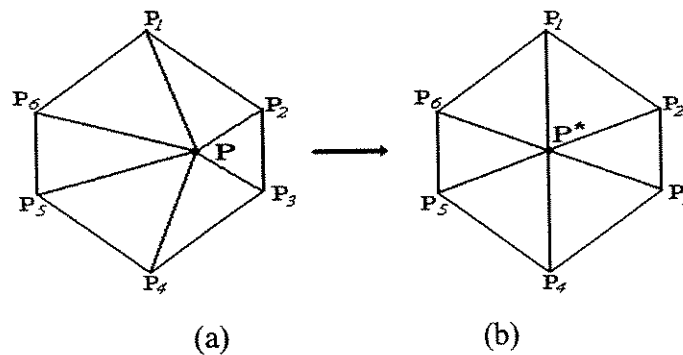


Figura 2.9 – Suavização, ou relaxação, da posição do nó P .

Com a ajuda desta técnica é possível eliminar as frentes de elementos indesejáveis, além de melhorar ainda mais o aspecto dos triângulos; mas, infelizmente, isto tem um alto custo computacional, o que torna o algoritmo consideravelmente mais lento à medida que aumentamos o número total de elementos.

2.3.2 – Técnicas de Refinamento pela Bissecção de Elementos

Esta técnica, inicialmente desenvolvida por Rivara [Rivara89], parte de uma lista de elementos a serem subdivididos, obtida de forma semelhante ao algoritmo mostrado na Seção 2.3.1. A diferença está na forma que é feita a divisão dos elementos. Enquanto na Seção 2.3.1 a divisão de um elemento, através de sua aresta de maior comprimento, era aplicada somente em casos especiais, especialmente quando o elemento era adjacente a

uma fronteira, neste método esta divisão é aplicada a todos elementos, estejam eles em uma fronteira ou não.

O processo de subdivisão consiste em identificar a maior aresta do elemento a ser particionado, e adicionar um novo nó no ponto médio desta aresta, no que é chamada bissecção generalizada. Quando a divisão é feita através de qualquer lado, é chamada de bissecção simples [Rivara97a]. Após serem feitas todas as divisões, inicia-se um processo para se identificar as situações de não conformidade dos elementos. O processo todo consiste de uma série de passos:

- (a) Partindo de uma malha inicial, Figura 2.10(a), subdividimos todos os elementos da lista através da bissecção generalizada.
- (b) A partir da malha subdividida, Figura 2.10(b), identificamos todos os pontos não conformes, no caso P e Q .
- (c) É feita uma bissecção simples do triângulo não conforme unindo-se o nó incompatível com o único vértice disponível. No nosso exemplo, dividimos o triângulo ACQ , conectando-se P e Q , e eliminando a não conformidade, conforme Figura 2.10(c).

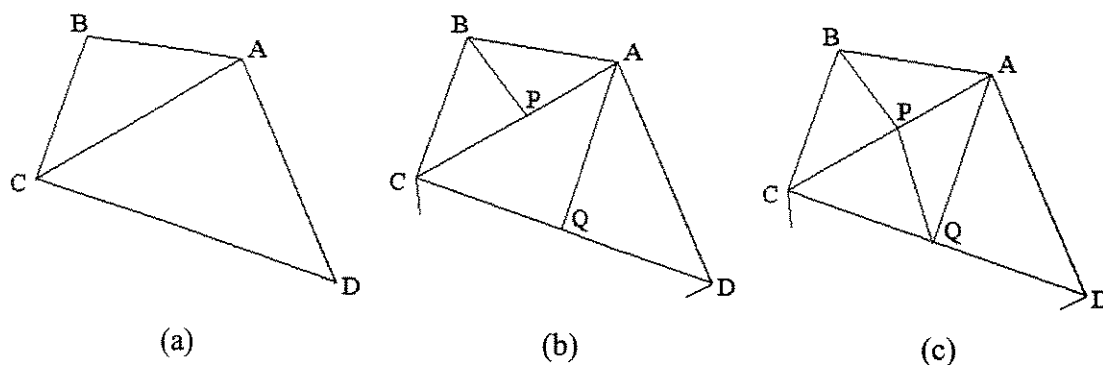


Figura 2.10 – Bissecção através do algoritmo apresentado por Rivara, em [Rivara89].

Após todas as divisões, é feita nova checagem para verificar se existem elementos acima do critério de parada. Caso existam, é feita nova lista e o processo todo é reiniciado, até que o critério de parada seja satisfeito por todos os elementos da malha.

Uma variante desta técnica foi proposta [Rivara89], na qual durante o processo de divisão, ao invés de se subdividir o elemento original em 2, a divisão gera 4 elementos, através da bissecção dos três lados do triângulo, conforme vemos na Figura 2.11. O restante do processo continua o mesmo para este caso.

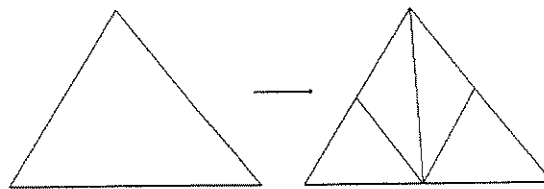


Figura 2.11 – Divisão de um elemento em 4 elementos.

Neste método não aparecem as ‘frentes’ de elementos e o aspecto da malha é bom, com elementos bem formados, ocorrendo uma transição suave entre os elementos maiores e os menores. As desvantagens deste processo consistem em: (a) o processo de corrigir os elementos não conformes consome muito tempo computacional, principalmente à medida que o número de elementos aumenta, e (b) a qualidade da malha final está fortemente vinculada à qualidade da malha inicial; assim, se a malha inicial tiver algum elemento ruim, este defeito não será corrigido como ocorria com a técnica baricêntrica.

Uma variação bastante robusta deste algoritmo foi, então, proposta para evitar o demorado processo de correção das não conformidades [Nambiar93]. Neste método, o elemento é dividido através da inserção de um nó em sua maior aresta e o elemento que compartilha esta aresta também é dividido através da conexão deste novo nó com o vértice oposto à maior aresta. Isto é feito automaticamente para evitar o surgimento de elementos não conformes.

Nesta técnica, a lista de elementos a serem subdivididos deve ser ordenada por ordem decrescente do tamanho da maior aresta do elemento, começando-se o processo a partir do elemento com a maior aresta, ou seja, a partir do primeiro item da lista. Assim, os elementos maiores são quebrados primeiro, o que evita o surgimento de elementos com um aspecto geométrico ruim e a deterioração da malha. Se a aresta a ser dividida através da bissecção generalizada estiver em uma fronteira externa, não temos nenhum problema, porém se ela pertencer a dois elementos, alguns cuidados devem ser tomados. Se a aresta em questão for a maior para ambos os elementos, então a divisão é feita simultaneamente nos dois elementos. Se esta condição não se verificar, o elemento com a aresta maior é adicionado ao início da lista, e o processo de divisão é reiniciado, até que alguma subdivisão seja efetuada. Na Figura 2.12 vemos uma exemplificação deste processo. Supondo que o elemento com maior aresta a ser dividido seja o elemento *A*, após a verificação vemos que o elemento *B* possui uma aresta maior que aquela que ambos compartilham e, assim, este elemento é colocado no final da lista para refinamento. Examinando os elementos *B* e *C*, vemos que ambos tem sua maior aresta em comum, sendo feita, então, a divisão, que criará 4 novos elementos, entre eles o elemento *D* que divide sua maior aresta com elemento *A*, inicialmente cotado para a divisão, e que é o próximo elemento da lista de subdivisão.

Neste método, os elementos resultantes da divisão são testados para verificar se estão acima da condição limite, caso em que serão adicionados à lista, sendo, então, necessária uma nova ordenação da lista. Este processo é feito repetidamente até que não reste nenhum elemento na lista.

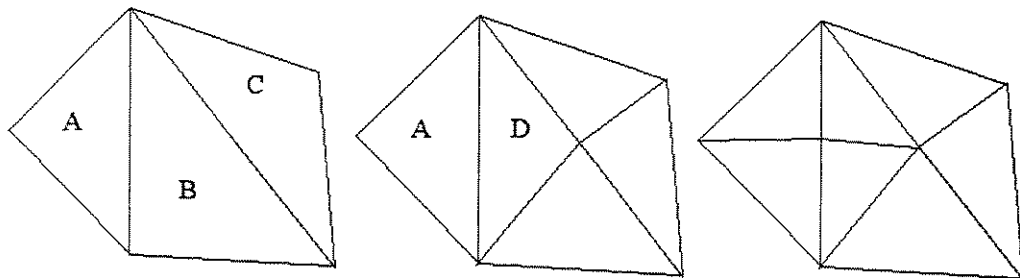


Figura 2.12 – Exemplificação do processo de bissecção pelo maior lado.

Como dissemos, este processo evita o custoso processo de busca por incompatibilidades, o que acelera bastante o processo, porém introduz a necessidade da manutenção de uma lista ordenada dos elementos, e que deve ser reordenada toda vez que um novo elemento é introduzido, o que se revela um problema, pois à medida que o número de elementos presentes na lista aumenta, o tempo de computação aumenta proporcionalmente. Outra desvantagem importante é que esta aproximação não resolve o problema da vinculação da qualidade final da malha em relação à malha inicial.

A divisão através da maior aresta de um elemento possui algumas propriedades, demonstradas em estudos anteriores [Rivara97a,b], a saber:

- O algoritmo de divisão através da maior aresta sempre termina em um número finito de passos, com a construção de uma triangulação conforme.
- Supondo que α é o menor ângulo interno da triangulação inicial, o uso iterativo do algoritmo irá produzir elementos cujos ângulos internos serão sempre iguais ou maiores a $\alpha/2$.
- Para qualquer triângulo t , a aplicação iterativa global do algoritmo, isto é, a divisão de todos os elementos da iteração anterior, cobre, de uma forma monotonicamente crescente, a área inicial de t com triângulos quase-equiláteros.

As duas últimas propriedades asseguram a construção de uma malha de boa qualidade, sem transições abruptas.

Recentemente, durante o processo de validação da nossa técnica (a ser apresentada na Seção 2.3.3), foi apresentado um algoritmo que contorna as necessidades de uma lista ordenada no algoritmo desenvolvido por Nambiar, e que era o mais robusto até o momento. Este novo algoritmo foi apresentado em 1997 por Rivara, nas referências [Rivara97a,b], e como foi apontado na Introdução, Seção 1.3, e como veremos na Seção a seguir, este algoritmo possui semelhanças marcantes com aquele por nós apresentado na mesma época

[Figueroa97], sendo, portanto, o resultado de desenvolvimento paralelo baseado nas mesmas premissas, e chegando a resultados semelhantes. O algoritmo de Rivara introduz o conceito de uma lista de elementos local, e não mais global como anteriormente. Este processo é denominado *LEPP – Longest Edge Path Propagation*, ou em uma tradução livre, Caminho de Propagação da Maior Aresta. Ele consiste, basicamente, em traçar a seqüência de elementos vizinhos que possuem ao menos uma aresta de maior comprimento do que a maior aresta pertencente ao triângulo precedente. Através das definições dadas por Rivara [Rivara97a], e algumas Figuras a seguir, tentaremos tornar mais claro este processo.

Definição 1: Para qualquer triângulo t_0 , de qualquer triangulação conforme T , o *LEPP* de t_0 será a lista ordenada de todos triângulos $t_0, t_1, \dots, t_{n-1}, t_n$, tal que t_i é o vizinho de t_{i-1} pela maior aresta de t_{i-1} , para $i = 1, 2, \dots, n$. Nós chamaremos isto como $LEPP(t_0)$.

Proposição 1: Para qualquer triângulo t_0 , de qualquer triangulação conforme, as seguintes propriedades são verdadeiras: a) para qualquer triângulo t , o $LEPP(t)$ será sempre finito; b) Os elementos $t_0, t_1, \dots, t_{n-1}$ estão em ordem crescente de maior aresta (para $n > 1$); c) Para o triângulo t_n do $LEPP$ de qualquer triângulo t_0 , podemos afirmar que ou (1) a maior aresta de t_n está em uma fronteira externa, e é superior à maior aresta de t_{n-1} , ou (2) t_n e t_{n-1} compartilham a maior aresta entre si.

Definição 2: Dois triângulos adjacentes (t, t^*) serão chamados de um par de triângulos terminais quando estes elementos compartilham a mesma maior aresta. Quando a maior aresta de um elemento t estiver em uma fronteira externa, chamamos este triângulo de triângulo terminal de fronteira.

Para ilustrar estas idéias vemos, na Figura 2.13(a), a triangulação original, e o $LEPP$ de t_0 , que corresponde à lista ordenada de triângulos $\{t_0, t_1, t_2, t_3\}$. Os elementos t_3 e t_2 são um par de triângulos terminais. Através da seqüência de figuras podemos acompanhar o funcionamento do algoritmo, até que o elemento t_0 seja subdividido e se obtenha a malha final, Figura 2.13(g). Na malha final numeramos os nós criados a cada passo, em ordem crescente.

Esta técnica permite a criação de elementos conformes e de bom aspecto geométrico, porém a qualidade final da malha ainda permanece fortemente condicionada à qualidade inicial da malha, pois novos nós são constantemente adicionados à malha, mas seu aspecto original permanece inalterado. Assim, se tivermos uma triangulação inicial ruim, a malha resultante permanecerá ruim.

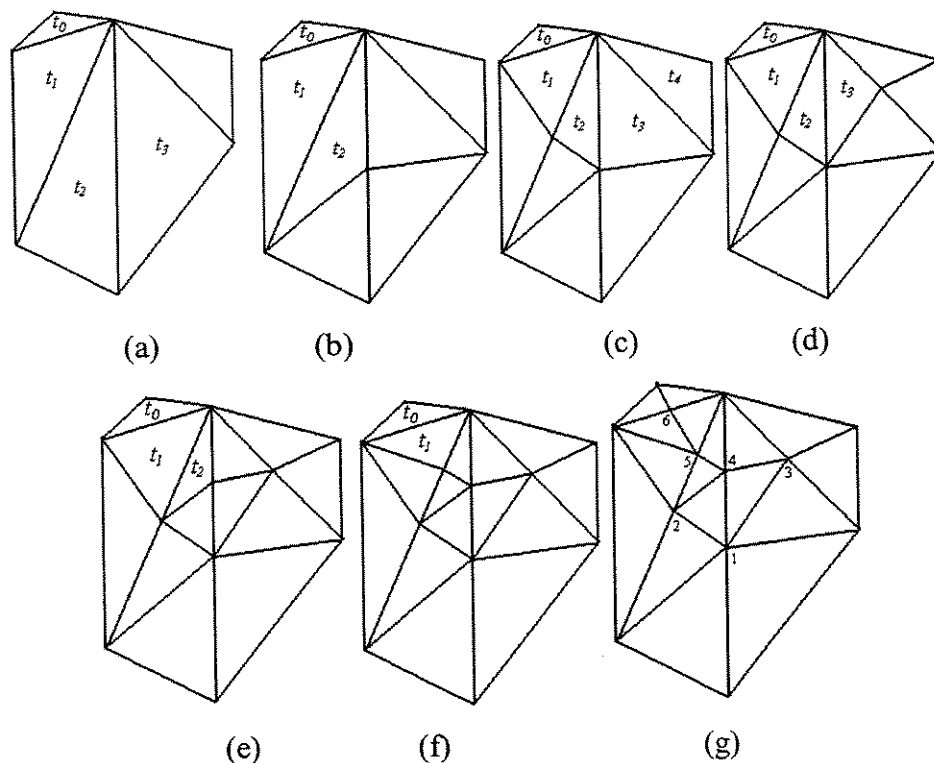


Figura 2.13 – Refinamento através da bissecção+LEPP. (a) Malha inicial, (b)-(f) passos do processo e (g) malha final.

Uma variação desta técnica utiliza a inserção de pontos de Delaunay em conjunto com o LEPP, para melhorar as malhas existentes. A autora ressalta que não se trata exatamente de um refinamento, pois o uso deste procedimento não irá necessariamente produzir um triângulo menor que o original, mas acarretará uma melhoria na qualidade da malha em relação à qualidade inicial, conforme [Rivara97b].

De forma semelhante ao algoritmo anterior, este método usa o conceito de LEPP para chegar à maior aresta e ,em seguida, fazer uma inserção de ponto de Delaunay nesta maior aresta. Esta inserção consiste basicamente em dois passos: a) Divide-se a maior aresta através do seu ponto médio, e b) efetua-se uma retriangulação entre todos os elementos que foram afetados pela inserção do novo nó para que satisfaçam ao teste da circunferência (Seção 2.3.1). Este procedimento é feito até que se chegue ao triângulo t_0 .

Na Figura 2.14, vemos um exemplo desta técnica, usando a mesma malha utilizada no exemplo anterior. Na Figura 2.14(a) vemos a malha inicial, aonde temos que $LEPP(t_0) = \{t_0, t_1, t_2, t_3\}$. As Figuras 2.14(b) até 2.14(d) ilustram o processo até se atingir o elemento t_0 , que é destruído na malha final, Figura 2.14(d).

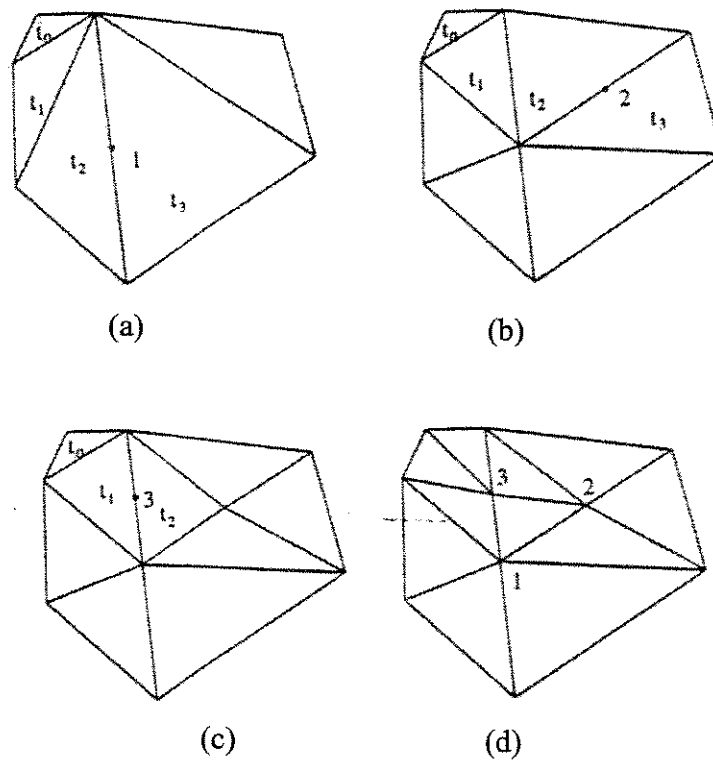


Figura 2.14 – Exemplo de refinamento com o segundo Algoritmo de Rivara.

Este algoritmo é o mais eficiente dentre os estudados até o momento. Entretanto, ele necessita de uma triangulação inicial que passe no teste da circunferência,

pois devido à melhora local dos elementos, eventuais triângulos ruins que não sejam afetados pelo algoritmo permanecerão com um aspecto ruim. Rivara também aponta que o tempo de execução deste algoritmo é linear, devido ao fato de que o custo de tempo para a inserção de um novo nó é constante [Rivara96].

2.3.3 – Técnica Proposta

A técnica que propomos consiste em uma conjunção dos algoritmos apresentados acima, preservando as suas qualidades e descartando as suas deficiências. A seguir, descrevemos a linha de raciocínio que nos levou a desenvolver um algoritmo semelhante àquele apresentado por Rivara, conforme discutido na Seção anterior.

2.3.3.1 – Desenvolvimento do Algoritmo

Como o objetivo deste trabalho era desenvolver um algoritmo para a geração eficiente de malhas de elementos finitos que fossem totalmente compatíveis e conformes, estudou-se quais vantagens cada técnica traria para nós. Como, na época, não havia publicações a respeito dos algoritmos de Rivara baseados no LEPP, estudamos as técnicas baricêntrica, descrita por Fernandez [Fernandez93], e da bissecção com uma lista ordenada, descrita por Nambiar [Nambiar93], e buscamos as deficiências e vantagens de cada técnica.

Ambas técnicas geram malhas conformes e totalmente compatíveis, e após uma análise cuidadosa, chegamos aos fatores de interesse de cada técnica, conforme detalhamos a seguir:

Técnica Baricêntrica:

Vantagens:

- Rapidez na geração da malha.

- Melhoria do aspecto da malha, devido ao uso das trocas de diagonais de Delaunay.

Desvantagens:

- Surgimento de transições abruptas entre elementos menores e maiores.
- Necessidade de um processo demorado de suavização para a correção destas transições.

Técnica de Bisseccão com lista ordenada:

Vantagens:

- Malhas de bom aspecto geométrico, sem transições abruptas.
- A aplicação sucessiva deste algoritmo tende a gerar triangulações quase-equiláteras.

Desvantagens:

- A qualidade da malha final está vinculada à qualidade da malha inicial.
- A necessidade da manutenção da lista ordenada de elementos para divisão, que consome muito tempo, principalmente para grandes quantidades de elementos.

Analisando a técnica de bissecção com ordenação dos elementos, notamos que a mesma produzia malhas sem transições abruptas. A nossa escolha inicial foi estudar uma forma de aperfeiçoar esta técnica. Como gostaríamos que a qualidade de nossa malha final fosse independente da qualidade da malha inicial, um dos problemas da técnica introduzida por Nambiar, o nosso objetivo era maximizar o ângulo mínimo, e isto era possível através da aplicação da troca de diagonais de Delaunay, como mostrado na Seção 2.3.1. Assim, chegou-se a um algoritmo no qual, após a bissecção do elemento, era aplicado o teste da circunferência a cada elemento resultante desta divisão.

Com isto chegou-se a um algoritmo que produzia triangulações bastante satisfatórias em termos de qualidade e aspecto geométrico, porém que ainda não satisfazia a necessidade de um algoritmo rápido, pois ainda era baseado no algoritmo de Nambiar, o qual faz uso de uma lista ordenada, pelo tamanho da maior aresta, de elementos para subdivisão. Então, o

próximo passo seria descobrir ou desenvolver uma técnica que contornasse a necessidade da lista ordenada.

Através de uma pesquisa na literatura, chegamos a uma técnica denominada visita por adjacência [Borouchaki96]. Nesta técnica, os elementos vizinhos a um determinado elemento eram visitados, até se chegar a um elemento base que serviria para a inserção de um novo nó em seu interior. Assim, utilizando este princípio, chegamos a uma nova aproximação para chegar a um elemento que seria a base para a bissecção.

Nesta aproximação, identificamos qual a maior aresta do elemento em análise e verificamos se o elemento vizinho, que compartilha esta aresta, não tem uma aresta maior, caso em que o mesmo critério é aplicado a este elemento. Isto é feito até se chegar ao elemento com a maior aresta a partir do elemento inicial, quando então se faz a bissecção deste elemento. A vantagem desta técnica em relação à lista ordenada, é que não há a necessidade de uma ordenação global dos elementos, já que através desta nova técnica só é preciso uma ordenação local, que já é feita pelo próprio algoritmo. Na Figura 2.15 vemos um exemplo de visita por adjacência, que ajuda a tornar mais claro o processo. O elemento 1 foi originalmente marcado para subdivisão, mas após verificar que o elemento vizinho (2) a ele possuía uma aresta maior que o elemento 1, passou-se, então, para o elemento 2. O processo se repete até que se chega ao elemento 4, no qual a sua maior aresta é a mesma que a do elemento 5, restando apenas passar para a subdivisão do elemento, através da bissecção na forma descrita na Seção 2.3.2.

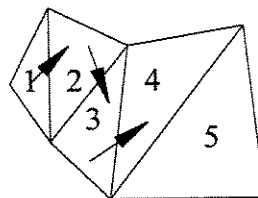


Figura 2.15 - Exemplo de visita por adjacência, para se achar o elemento com a maior aresta em determinada região.

Como vimos anteriormente, este processo foi desenvolvido de forma paralela por Rivara e apresentado pela mesma nas referências [Rivara97a,b], aonde é chamado de *Caminho de Propagação da Maior Aresta* ou LEPP, Long Edge Path Propagation, e aparentemente foi baseado em premissas equivalentes àsquelas utilizadas para o desenvolvimento de nosso algoritmo. Devido à publicação de seus resultados ser bem aceita pela comunidade científica, utilizaremos aqui a terminologia utilizada por ela em seus trabalhos.

Porém nosso algoritmo ainda falhava em um ponto: Não maximizava totalmente o ângulo mínimo da malha nos locais aonde o processo de troca de diagonais não atingia, como, por exemplo, regiões próximas a interfaces. Para contornar esta deficiência, acrescentou-se um último passo ao processo, que também se diferencia do algoritmo de Rivara, no qual as posições dos nós da malha são suavizados uma única vez, de acordo com a Equação 2.13. Este procedimento melhora a qualidade e aumenta o ângulo mínimo da malha, sem comprometer a velocidade do algoritmo, uma vez que o processo de otimização é feito somente uma vez, e não recursivamente até se atingir uma estabilização, que pode levar mais de 10 iterações, como ocorre no algoritmo de refinamento baricêntrico, Seção 2.3.1.

2.3.3.2 – Implementação final do algoritmo

Nesta Seção apresentamos o algoritmo em sua forma final. Na Figura 2.16 apresentamos um fluxograma do mesmo, e, a seguir, detalharemos cada passo de seu funcionamento.

- 1 – Primeiramente, aplicamos o teste da circunferência na malha inicial, para corrigir eventuais triangulações ruins que possam existir.
- 2 – Após isto, criamos uma lista de elementos a serem subdivididos, isto é, aqueles elementos que estão acima de certa condição limite, que pode ser uma condição *a priori* ou *a posteriori*, conforme vimos na Seção 2.2.

- 3 – Inicia-se o laço principal através da escolha do primeiro elemento da lista para subdivisão.
- 4 – É feito o teste da maior aresta, para definir se o elemento em questão é um triângulo terminal, ou se seu vizinho possui uma aresta maior. Caso o elemento seja realmente um triângulo terminal, passamos para o passo 5, caso contrário colocamos o elemento vizinho na primeira posição da lista para refinamento, e retornamos ao passo 2. Se este elemento já estiver na lista inicial, devemos proceder a uma atualização da mesma, para que o elemento em questão não seja considerado duas vezes.
- 5 – Antes de fazermos a subdivisão, é necessário verificar se a aresta a ser dividida está em uma fronteira externa, ou se ela é compartilhada por dois elementos. Neste último caso, verificamos se o outro triângulo também está listado para subdivisão e excluimos ambos (o elemento original e seu vizinho, que serão divididos juntos no processo) da lista de elementos.
- 6 – Fazemos a subdivisão do elemento através da inserção de um novo nó no ponto médio da maior aresta, criando dois novos elementos para triângulos terminais de fronteira ou quatro, quando tratar-se de um par de triângulos terminais.
- 7 – É aplicado o teste da circunferência aos elementos resultantes da subdivisão.
- 8 – Verificamos se estes elementos estão acima da condição limite especificada no passo 2, caso em que adicionamos estes elementos à lista de elementos para subdivisão.
- 9 – Caso não tenhamos atingido o final da lista, retornamos ao passo 3, senão passamos ao passo seguinte.
- 10 – Após chegarmos ao final da lista, aplicamos a troca de diagonais de Delaunay a todos elementos da malha a fim de corrigir eventuais distorções e melhorar o aspecto da malha.
- 11 – Caso tenhamos atingido o critério de parada escolhido (*a priori* ou *a posteriori*) terminamos o programa, senão retornamos ao passo 2. Isto é feito até que a condição de parada seja satisfeita, ou até que não exista nenhum elemento acima do critério limite.
- 12 – Após terminado todo o processo de refinamento, é feita uma otimização da malha, através da suavização da posição dos nós, uma única vez.

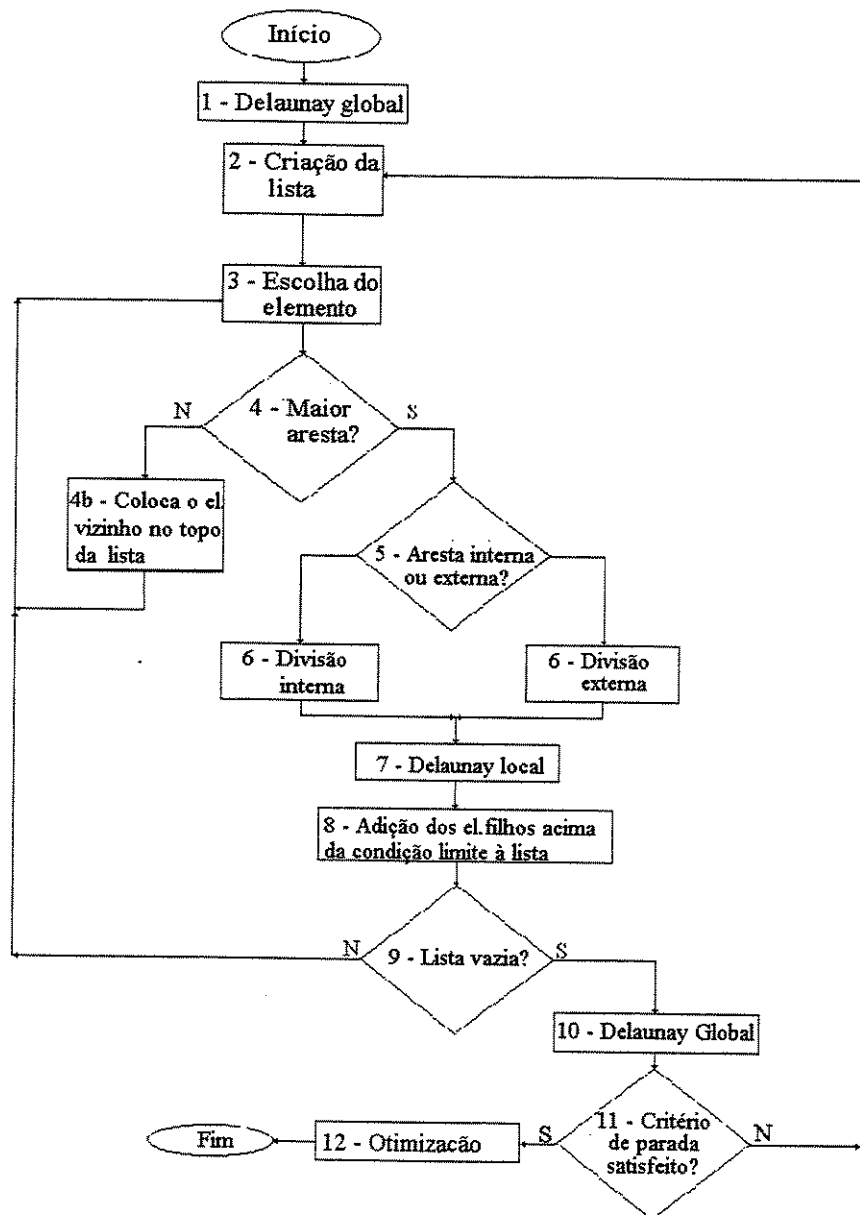


Figura 2.16 - Fluxograma do algoritmo de refinamento.

Uma das diferenças em relação ao trabalho de Rivara, é a possibilidade de se acelerar o processo de busca do triângulo terminal do LEPP, através de um correto uso da nossa estrutura de dados. Como pode ser visto em detalhes no Apêndice A, nossos dados estão organizados em pontos e elementos vizinhos, sendo que os dois primeiros pontos representam a maior aresta, que pertence também ao primeiro elemento vizinho da nossa estrutura de dados. Assim, ao invés de se passar de elemento a elemento como no processo

de visita por adjacência descrito acima, é possível acessar diretamente a informação do elemento sucessivo ao elemento vizinho no LEPP. Ainda é necessário o acesso ao elemento vizinho, mas isto se dá de forma mais rápida do que ocorre na visita por adjacência simples. Através da Figura 2.17 tentaremos deixar mais claro o processo. Da mesma forma que anteriormente, o elemento 1 foi marcado para divisão, e verificou-se que o elemento 2 possui uma aresta ainda maior. Através da nossa estrutura de dados, achamos qual o elemento que compartilha esta aresta com o elemento 2 e chegamos ao elemento 3. Observe-se que foi eliminado um passo do laço $3 \rightarrow 4 \rightarrow 4b$. O processo é repetido novamente, e chegamos ao elemento 5, que forma um par de triângulos terminais com o elemento 4. Como vemos, foram necessárias 2 operações para se atingir o par de triângulos terminais, e não 3, como no caso anterior.

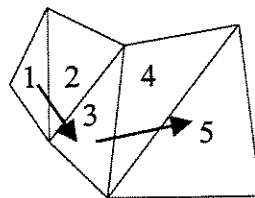


Figura 2.17 - Exemplo de visita por adjacência acelerada.

No fluxograma mostrado na Figura 2.16, e na explicação do mesmo, utilizamos o processo de visita por adjacência simples. Caso utilizemos o processo de visita por adjacência acelerada, uma pequena modificação deve ser feita no passo 4. Ao invés de colocarmos o vizinho do elemento original no topo da lista, colocaremos o elemento que compartilha a maior aresta com este elemento vizinho.

Como vimos com detalhes, esta técnica possui vantagens em relação às suas antecessoras, em alguns pontos: é mais rápida e produz elementos melhores que a técnica da bissecção apresentada por Nambiar [Nambiar93], e elimina as ‘frentes’ de elementos, sem a necessidade do uso recursivo de algoritmos de suavização, utilizados no algoritmo de Fernandez [Fernandez93]. Possui, ainda, algumas vantagens em relação aos algoritmos desenvolvidos por Rivara, que utilizam o LEPP.

Para validar nosso algoritmo, uma série de testes foi efetuada, inclusive comparando-se as malhas obtidas pelo nosso programa com aquelas obtidas utilizando-se os diferentes algoritmos aqui apresentados. Tais testes são apresentados na Seção 6.2, Resultados do Refinamento Bidimensional.

Capítulo 3

Malhas Tridimensionais Superficiais

No Capítulo anterior, apresentamos os diversos aspectos envolvidos no refinamento adaptativo de malhas de elementos finitos triangulares (técnicas de partição, métodos de controle do refinamento, etc.). Neste Capítulo, apresentaremos o refinamento superficial tridimensional e seus diferentes aspectos. Como veremos adiante, boa parte do método utilizado no Capítulo 2 pode ser utilizado aqui, porém algumas modificações serão necessárias, principalmente quanto à localização dos nós resultantes da subdivisão de triângulos e quanto à troca de diagonais de Delaunay. Apresentaremos, ainda, uma adaptação do algoritmo apresentado no Capítulo anterior para o refinamento superficial tridimensional.

O uso de malhas tridimensionais superficiais tem sido bastante extenso, compreendendo desde a animação por computador até a análise de estruturas mais complexas, principalmente na área de Engenharia Mecânica [Peraire88, Hinton91, Bajaj96,

Blacker96, Canann97, Thompson99]. Por outro lado, sua utilização em problemas eletromagnéticos tem sido relativamente recente [Jin93, Itoh96, Renhart96, Morcillo98]. Um dos motivos para o estudo da geração de malhas tridimensionais superficiais, é que esta malha pode ser utilizada como um primeiro passo para a geração de malhas tridimensionais volumétricas, através de técnicas como ‘Advancing Front’ [Tristano98, Thompson99].

3.1 – Representação das Superfícies

A construção de malhas de elementos finitos em superfícies tridimensionais é um dos mais importantes aspectos da simulação de estruturas em três dimensões. As malhas 3D superficiais devem atender aos mesmos requisitos de conformidade, compatibilidade, qualidade e suavidade exigidos das malhas bidimensionais, e, além disso, ainda devem se adaptar às características específicas de cada superfície, incluindo linhas de interseção, curvaturas e estruturas complexas.

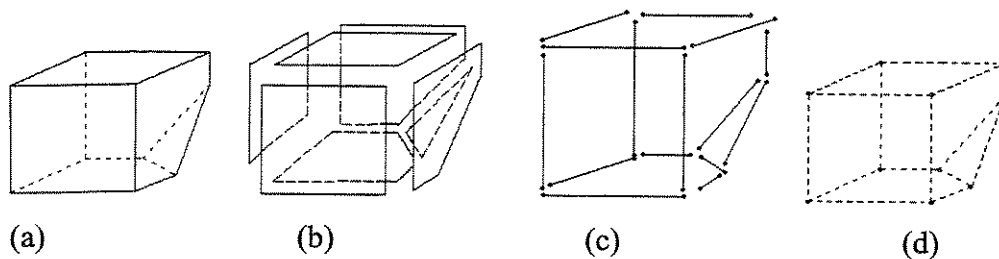


Figura 3.1 - Decomposição de um modelo 3D em seus componentes geométricos.

Geralmente, o domínio a ser discretizado, que também pode ser chamado de *domínio computacional*, pode ser representado como um conjunto de superfícies que, quando devidamente conectadas entre si, descrevem o objeto tridimensional com exatidão. Isto é conhecido como representação por fronteira, ou ‘*B-rep*’ (do inglês *Boundary representation*), e é utilizada pela maioria dos programas comerciais e de pesquisa.

Em uma representação de contorno, o domínio computacional pode ser considerado como um poliedro generalizado que é a união de um conjunto de superfícies, ligadas através de linhas, por sua vez conectadas através de vértices [Peraire88], conforme vemos nas Figuras 3.1. Na Figura 3.1(a) mostramos o domínio tridimensional e na Figura 3.1(b) mostramos as superfícies que o compõem. Na Figura 3.1(c) apresentamos as linhas que formam as superfícies e finalmente na Figura 3.1(d) apresentamos os vértices que compõem as linhas.

Assim, dados geométricos a respeito da forma das superfícies e linhas devem ser adicionados à estrutura de dados utilizada anteriormente para o caso 2D, ver Apêndice A. Também devem ser armazenadas informações a respeito da conectividade das superfícies entre si [Souza93]. Isto é necessário para evitar divisões não-conformes de triângulos, que poderiam ocorrer durante o refinamento de dois triângulos que compartilham a mesma aresta, mas que estão em superfícies diferentes, conforme Figura 3.2.

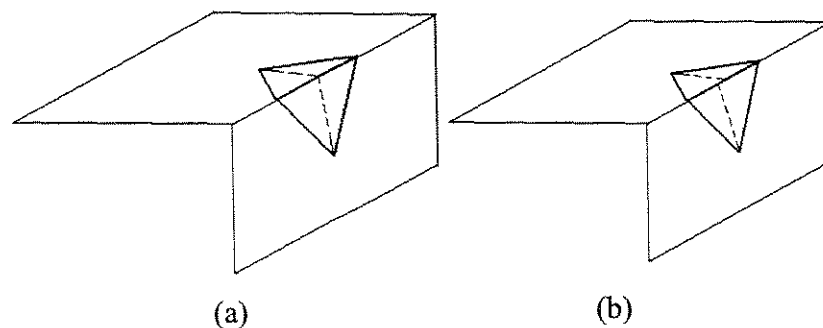


Figura 3.2 – Refinamento (a) conforme e (b) não-conforme, em superfícies tridimensionais.

Nos exemplos utilizados acima, vemos apenas estruturas planas e linhas retas, o que facilita o trabalho de refinamento de cada superfícies, e serve de base para o refinamento de superfícies mais complexas. Na maioria dos casos, temos de trabalhar com superfícies e linhas curvas. Nestes casos uma aproximação, usualmente utilizada, é a representação destas superfícies e linhas através de coordenadas paramétricas [Hinton91, Thompson99]. Esta representação paramétrica é facilmente obtida e proporciona uma descrição acurada

das mesmas em termos das coordenadas parametrizadas, além de facilitar a discretização da superfície.

3.2 – Modelamento das Superfícies

As superfícies podem ser representadas em termos da informação fornecida pelos contornos da superfície e por funções auxiliares das coordenadas paramétricas (u, v) , que variam entre 0 e 1, ao longo dos contornos relevantes das superfícies [Hinton91, Tristano98]. A posição do vetor $\mathbf{r}$, em relação a um ponto na superfície, é dada por

$$\mathbf{r}(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} u^i v^j \quad (3.1)$$

aonde a_{ij} é um coeficiente que contém uma combinação linear da posição e vetores (tangente e normal) dos vértices das superfícies, e u^i e v^j são as coordenadas paramétricas associadas a cada eixo coordenado. Na Figura 3.3, vemos um exemplo do mapeamento de um retângulo 3D em um espaço paramétrico 2D, no qual apontamos o vetor $\mathbf{r}$.

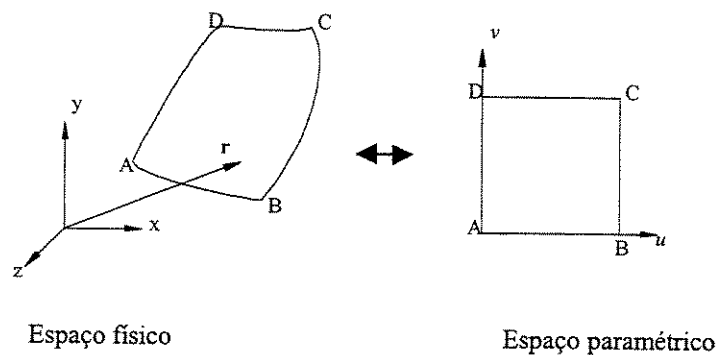


Figura 3.3 - Mapeamento de um retângulo tridimensional no espaço paramétrico bidimensional e vice-versa.

Uma vez que os parâmetros da malha no plano paramétrico tenham sido determinados, o refinamento da malha é conduzido. Sabendo as posições dos pontos nodais no plano paramétrico, as posições correspondentes dos pontos nodais na superfície 3D podem ser determinados usando a equação mostrada acima. A conectividade dos elementos entre si permanece idêntica em ambos espaços, devido à correspondência entre ambos os planos.

Quando tratamos de superfícies curvas simétricas, tais como esferas, cilindros, parábolas, elipses, etc., algumas simplificações podem ser feitas em coordenadas angulares. Analisemos os casos de duas superfícies bastante simples: uma esfera e um cilindro. As coordenadas cartesianas de cada uma das formas acima devem satisfazer as seguintes equações:

$$\text{Esfera: } x^2 + y^2 + z^2 = a^2; \quad (3.2a)$$

$$\text{Cilindro: } \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1; \quad (3.2b)$$

onde a Equação (3.2b) corresponde a um cilindro com eixo coincidente ao eixo z do sistema de coordenadas cartesiano. Caso tenhamos a igual a b , então teremos um cilindro circular de raio igual a a , e a Equação (3.2b) fica igual a $x^2 + y^2 = a^2$.

De posse destes dados, isto é, dos fatores de cada equação, é perfeitamente possível definir estas superfícies no espaço. A transformação destas equações no espaço real, para as equações no espaço paramétrico, é feita de um modo tal que sua descrição cartesiana através de três coordenadas (x,y,z) seja feita, agora, através de duas coordenadas angulares (u,v) . Após

$$\text{Esfera} = \begin{cases} u = \arctan(x/y) \\ v = \arccos(z/\sqrt{x^2 + y^2 + z^2}) \end{cases} \quad (3.3a)$$

$$\text{Cilindro Circular} = \begin{cases} u = \arctan(x/y) \\ v = z \end{cases} \quad (3.3b)$$

Assim, após transformarmos todos os dados de nossa malha inicial em coordenadas cartesianas para coordenadas angulares, podemos dar início ao refinamento da mesma. Isto pode ser feito através de uma das técnicas ou algoritmos discutidos na Seção 2.3. Quando obtivermos a malha desejada, o próximo passo será fazermos a inversão da transformação e conseguirmos o mapeamento dos nós da malha na superfície pretendida. Esta inversão é feita facilmente através das equações a seguir:

$$\text{Esfera} = \begin{cases} x = a \cdot \text{sen}(u) \cdot \cos(v) \\ y = a \cdot \text{sen}(u) \cdot \text{sen}(v) \\ z = a \cdot \cos(u) \end{cases} \quad (3.4a)$$

$$\text{Cilindro Circular} = \begin{cases} x = a \cdot \cos(u) \\ y = a \cdot \text{sen}(u) \\ z = z \end{cases} \quad (3.4b)$$

Nesta sequência de transformações, é necessário tomar cuidado para que cada ponto esteja corretamente mapeado no seu quadrante específico, para evitar a perda de informação que pode facilmente ocorrer.

O mesmo procedimento utilizado acima, para superfícies curvas regulares, pode ser utilizado para superfícies de revolução, tais como cilindros e cones circulares. Para isso, consideramos que a linha de base, o eixo ao redor do qual é feita a revolução, está em um ângulo igual a zero. Isso significa que, por exemplo, o raio será constante ao longo de determinado eixo para o caso de um cilindro circular, e que será uma função do tipo $y = ax$ para o caso cônico. Assim, a variação do raio ao longo de determinado eixo é definida através de funções polinomiais, que ficam mais complexas à medida que as nossas superfícies de revolução também se tornam mais elaboradas.

3.3 – Troca de Diagonais em Malhas Superficiais

É possível utilizarmos do critério da circunferência de Delaunay para melhorar o aspecto de nossa malha tridimensional superficial, porém existem alguns empecilhos em sua aplicação. O primeiro, é a dificuldade em se mapear corretamente o centro do círculo circunscrito a um triângulo em uma superfície curva. Também se torna difícil verificar se determinado ponto de outro triângulo está, ou não, dentro de tal círculo circunscrito diretamente no espaço cartesiano [Chew93]. Por exemplo, na Figura 3.4, a troca de diagonais não é recomendada pois devido à grande curvatura da superfície, com a troca de diagonais, a aresta que separa os elementos não estará mais sobre a superfície, apesar do formato dos triângulos aparentemente ter melhorado.

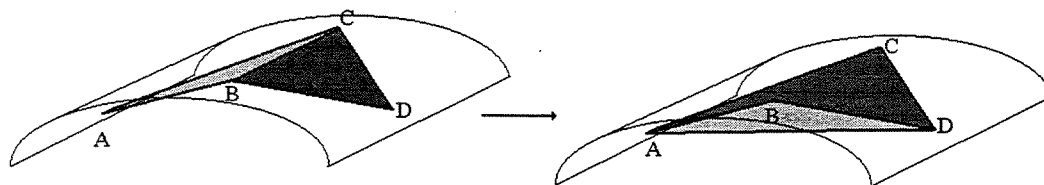


Figura 3.4 – Exemplo de troca de diagonais imprópria.

Uma das aproximações sugeridas consiste em se fazer a troca de diagonais no espaço paramétrico, mas, para isso, é necessário se fazer um ajuste na equação dos círculos, pois devido à distorção que ocorre no espaço parametrizado, nem sempre um círculo circunscrito a um triângulo no espaço real corresponde a um círculo no espaço paramétrico. Em [Chen97], é proposta a aproximação deste círculo através de uma elipse, que é então utilizada para se fazer a checagem se é necessária a troca de diagonais. Entretanto, isto só se torna útil quando lidamos com superfícies muito curvas. Em superfícies de menor curvatura, é possível fazer-se uma aproximação mais rápida através da simples projeção dos triângulos em um plano bidimensional e pela utilização do critério de troca das diagonais de Delaunay da forma usual. Através da Figura 3.5 vemos um exemplo deste procedimento, no qual os triângulos da superfície S são projetados em um plano, no qual é

feito o teste da circunferência e a troca de diagonais de Delaunay. A seguir, os novos triângulos são mapeados de volta à superfície curva S . Isto só é possível devido à pequena curvatura da superfície S , o que implica em uma baixa distorção nas transformações.

A medida de qualidade geral da malha e dos elementos isolados, é feita de maneira semelhante à apresentada na Seção 2.1, sendo necessárias apenas pequenas mudanças para levar em conta a característica tridimensional.

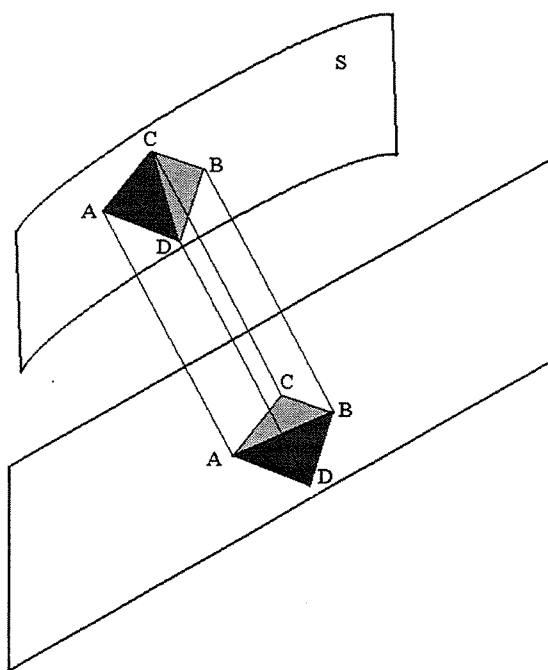


Figura 3.5 – Troca de diagonais através da projeção.

3.4 – Implementação do Algoritmo de Refinamento

Como discutimos nas Seções anteriores, para que possamos utilizar o nosso algoritmo bidimensional, visto na Seção 2.3.3, são necessárias algumas modificações. A

primeira delas diz respeito à estrutura de dados. Foram necessários o acréscimo de dois novos dados: o primeiro é a atualização das coordenadas, pois antes necessitávamos apenas das coordenadas x e y , e, agora, precisaríamos de uma nova coordenada, z ; o segundo dado diz respeito à definição de que tipo de superfícies estamos trabalhando (plano, esfera, etc.), e que elementos pertencem a cada uma. Para maiores detalhes, ver Apêndice A.

Outra mudança necessária estava na forma de cálculo do novo ponto a ser criado na aresta de maior tamanho. Utilizando a transformação de coordenadas, descrita na Seção 3.2, foi possível chegar a uma rotina que faz a subdivisão de uma forma segura, tal que cada novo ponto gerado pertence à superfície em refinamento, sem a criação de elementos não conformes e incompatibilidades. Podemos definir esta rotina por uma série de passos:

- 1 – Transformação de coordenadas cartesianas em coordenadas angulares: Utilizando os dados das superfícies, calculamos as coordenadas paramétricas dos vértices da maior aresta.
- 2 – Subdivisão: Utilizando os dados do espaço paramétrico calculamos as coordenadas do novo nó situado no ponto médio dos vértices.
- 3 – Transformação das coordenadas paramétricas em coordenadas cartesianas: Resgatamos os dados dos vértices e calculamos as coordenadas cartesianas do novo ponto criado no passo 2, através de seus dados.

A mudança final diz respeito ao critério de troca de diagonais de Delaunay: o teste da circunferência. Como vimos anteriormente, Seção 3.3, a determinação do centro do círculo circunscrito a um dado elemento não é uma matéria trivial e, por isso, optou-se por utilizar a projeção dos triângulos no plano xy . Optou-se por esta projeção para aproveitar o algoritmo já disponível para o caso bidimensional. Para se fazer esta projeção, são necessárias algumas transformações geométricas facilmente obtidas [Foley96]. O processo todo é feito da seguinte forma: inicialmente verificamos se o ângulo entre os dois triângulos permite que se faça esta projeção com mínimo de erro possível. Se o ângulo entre eles for suficientemente pequeno, então passamos às transformações necessárias.

Sabendo que três pontos no espaço definem um plano, podemos dizer que cada triângulo irá definir um plano. Achando o vetor unitário normal a cada plano e calculando a

soma entre as normais, acharemos um vetor v_o que é ortogonal à maior aresta. Agora basta acharmos o ângulo entre o vetor v_o e o eixo z , e rotacionarmos o conjunto todo de forma que o vetor v_o e o eixo z fiquem paralelos. Como a nossa condição inicial para o início do processo era de que os ângulos entre os dois elementos era suficientemente pequeno para não incorrer em distorções, basta supor que todos os nós estão situados no plano xy e aplicamos o teste da circunferência de Delaunay aos triângulos da mesma maneira que utilizamos no caso bidimensional. Para dar maior clareza, na Figura 3.6 mostramos os componentes desta explicação. Na Figura 3.6(a) temos os elementos aos quais queremos aplicar o teste de diagonais, projetados no plano xy , enquanto que na Figura 3.6(b) vemos os mesmos elementos agora projetados no plano xz . Nesta figura vemos, também, as normais a cada um dos elementos e o vetor v_o . Na figura seguinte, Figura 3.6(c), vemos novamente o vetor v_o , o eixo z , e o ângulo θ entre eles. Este é o ângulo que deve ser rotacionado para que os elementos E_1 e E_2 possam ser projetados com o menor erro possível, e o critério da circunferência de Delaunay possa ser aplicado.

Apesar destas transformações parecerem extremamente complicadas e demoradas, na prática elas são quase automáticas e não consomem muito tempo de computação, e como todas estas transformações (exceto a troca de diagonais) ocorrem de forma local, não é necessário fazermos a transformação inversa.

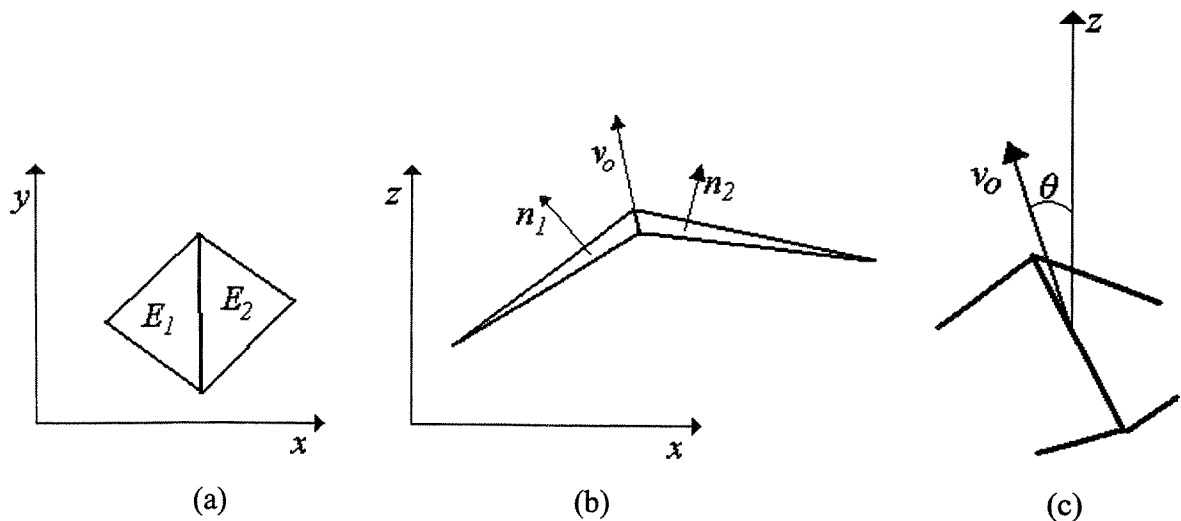


Figura 3.6 – Exemplo do processo necessário para troca de diagonais

Em superfícies simples (quadrados, por exemplo) a maioria destes triângulos são coplanares, implicando que ambos os vetores normais serão iguais. Com isso, não teremos a necessidade de calcularmos o vetor v_o , poupando um pouco de trabalho.

Também foi necessário eliminar-se a suavização final do algoritmo de refinamento superficial tridimensional. Como existem casos em que esta otimização poderia destruir a superfície que queremos modelar, como por exemplo em superfícies curvas, decidiu-se suprimir esta última tarefa no nosso algoritmo. Estão em estudo técnicas para adicionar esta funcionalidade ao nosso algoritmo.

Após feitas todas as modificações, chegamos ao algoritmo final, que é muito semelhante àquele apresentado na Seção 2.3.3.2. As modificações se encontram em três pontos: leitura de dados, divisão interna e externa e troca de diagonais. Alguns testes foram feitos, utilizando nosso algoritmo de refinamento, e os resultados são apresentados na Seção 6.3.

Capítulo 4

Malhas Tridimensionais Volumétricas

Nos Capítulos anteriores desenvolvemos um algoritmo para o refinamento bidimensional de malhas de elementos finitos triangulares e fizemos sua adaptação para o uso em superfícies tridimensionais, além de estudar as diversas características de cada caso, suas facilidades e dificuldades. A seguir, estudaremos as malhas tridimensionais volumétricas, ou malhas tetraédricas.

É errôneo tentar traçar um paralelo entre as técnicas de refinamento bidimensional e as técnicas envolvendo três dimensões. Um exame preciso do problema nos leva a algumas questões que são específicas do problema tridimensional, sem que haja correspondência em duas dimensões. Assim, uma regra facilmente aplicável a triângulos pode não ser verificável em tetraedros [George99]. Para ilustrar isto, citaremos dois exemplos que podem afetar vários aspectos do refinamento de uma malha 3D:

1. Um triângulo com todos lados iguais é necessariamente um triângulo equilátero, enquanto que um tetraedro com a mesma propriedade pode possuir uma forma

geométrica pobre e ter seu volume tão pequeno quanto desejarmos (ver definição de lascas nesta seção, por exemplo).

2. A área de um triângulo tende a zero somente se o comprimento de uma das arestas também tender a zero. Para os tetraedros isto não é verdade pois o volume de um tetraedro pode tender a zero enquanto suas arestas permanecem grandes (ver tetraedros do tipo ‘lasca’ na seção 4.1).

Portanto, algumas mudanças serão imprescindíveis, tanto na forma da divisão como na estrutura de dados. Veremos que é possível utilizarmos as técnicas estudadas nos Capítulos antecedentes, devidamente adaptadas, para construirmos um algoritmo que faça o refinamento destas malhas tetraédricas.

A geração e refinamento de uma malha de elementos finitos podem absorver muito tempo de análise, e isto se torna consideravelmente pior quando tratamos de elementos tetraédricos. As dificuldades principais estão principalmente relacionadas com a dificuldade em se descrever a geometria do problema de forma clara, e em visualizar a estrutura tridimensional, e a malha correspondente, em uma tela de computador bidimensional. Por isso, alguns pesquisadores preferem utilizar hexaedros em suas malhas, uma vez que elementos cúbicos são muito mais fáceis de trabalhar do que elementos tetraédricos [Golias94]. Entretanto, como o tetraedro é uma forma volumétrica simples, seu uso permite uma melhor descrição do domínio computacional do problema, não importando a complexidade do mesmo, o que o torna atrativo. Além disso, as funções polinomiais de expansão são facilmente definidas para estes elementos.

Como estudamos anteriormente, nosso algoritmo se baseia no refinamento adaptativo de uma dada malha inicial. Esta malha pode ser obtida através de diversos métodos, desde a sua criação totalmente manual até processos que geram a malha a partir dos dados a respeito do contorno da estrutura [Gonzalez96]. Estes métodos de geração não se constituem no objetivo da tese, e portanto não serão tratados aqui. Nos ateremos somente ao estudo dos métodos de refinamento para malhas volumétricas.

4.1 – Refinamento Tridimensional

Da mesma forma que no refinamento bidimensional, a qualidade de um elemento tridimensional é muito importante, embora agora a qualidade máxima seja mais difícil de se obter. A fórmula para se obter a qualidade de um tetraedro é dada por

$$Q = 3 \cdot \frac{r}{R} \quad (4.1)$$

isto é, a qualidade Q de um elemento é igual a três vezes a razão entre o raio da esfera inscrita r e o raio da esfera circunscrita R . Uma fórmula computacionalmente mais eficiente (em torno de 20 vezes), e que produz os mesmos resultados, é dada por

$$Q = \frac{12}{\sqrt{2}} \frac{V}{L_{max}^3} \quad (4.2)$$

na qual V é o volume do tetraedro e L_{max} é o comprimento de sua maior aresta [Golias94]. Um tetraedro com todos os lados e ângulos iguais entre si terá uma qualidade máxima, que será igual a 1, enquanto que os elementos degenerados, ver figura 4.1, possuirão valores de qualidade próximos de zero.

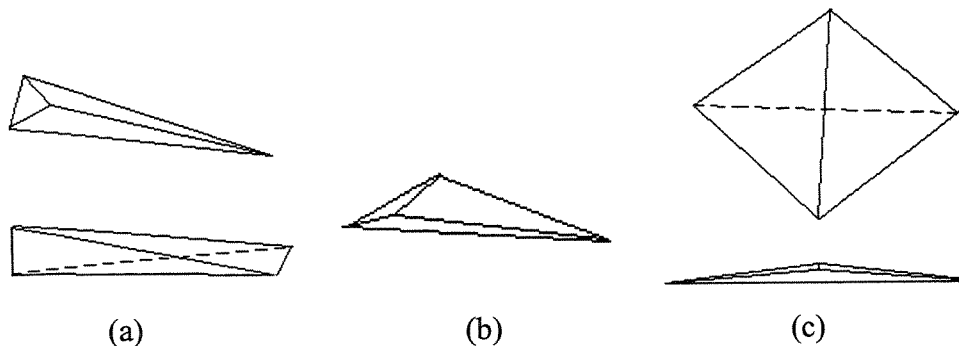


Figura 4.1 – Exemplos de elementos degenerados. (a) Cunhas, (b) cume e (c) lascas.

Estes elementos degenerados devem ser evitados, tanto na construção da malha quanto no seu refinamento. Os elementos degenerados do tipo ‘lasca’ são particularmente indesejáveis. Uma descrição mais exata destes elementos os define como o arranjo de 4 vértices, igualmente espaçados, ao redor do equador de uma esfera, aonde um dos vértices está levemente perturbado em relação à latitude central, como vemos na Figura 4.2 [Shewchuk97]. Devido à sua aparente igualdade entre ângulos internos e arestas, se fizermos uma analogia com o caso bidimensional seremos induzidos a pensar que o elemento é bem formado, o que não é verdade, pois se aplicarmos o teste de qualidade, Equação 4.2, o volume do tetraedro será pequeno, implicando em um baixo índice de qualidade.

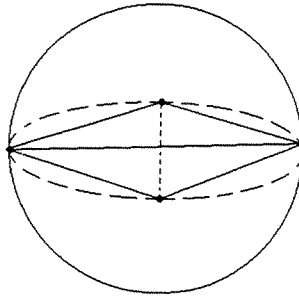


Figura 4.2 – Um tetraedro degenerado do tipo lasca.

O surgimento destes elementos degenerados pode ser minimizado através de técnicas de transformação topológica, aplicadas durante o refinamento e durante a fase de estabilização da malha, técnicas estas que serão descritas mais adiante, na Seção 4.2.

Como vimos anteriormente, o nosso algoritmo se baseia na inserção de novos nós em uma malha inicial pré-existente. Enquanto no caso bidimensional existiam apenas duas formas de inserção (na aresta e no baricentro), em elementos tetraédricos existem três formas de se fazer a inserção de um novo nó no elemento: no interior do tetraedro, na sua face e na sua aresta, dividindo este elemento em quatro, três e dois tetraedros respectivamente, conforme vemos nas Figuras 4.3. Na Figura 4.3(a) temos a inserção do nó no interior do tetraedro, gerando 4 elementos: $ABDE$, $ACDE$, $ABCE$ e $BCDE$. Na Figura 4.3(b) vemos a inserção do ponto nodal na face ABD , gerando 3 elementos: $ABCE$, $ACDE$ e $BCDE$, e, finalmente, na Figura 4.3(c) vemos a inserção do nó E na aresta AD , gerando 2

elementos: $ABCE$ e $BCDE$. Naturalmente, nos casos em que a aresta ou face é compartilhada por outro(s) elementos, este(s) também deve(m) ser subdividido(s).

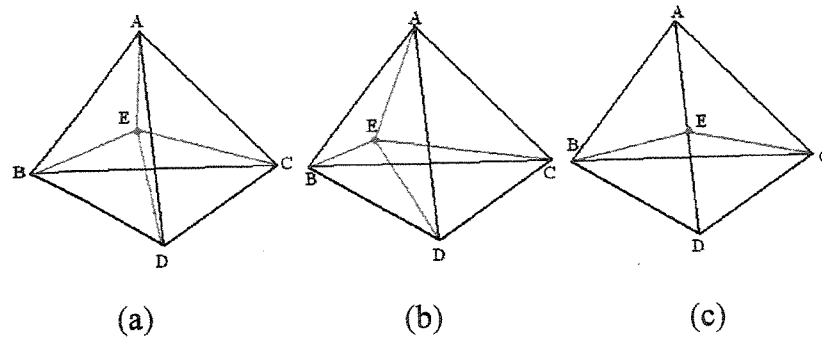


Figura 4.3 – Formas de inserir um novo nó E no tetraedro formado pelos nós $ABCD$. As linhas pretas representam as arestas do tetraedro original e as linhas vermelhas as arestas resultantes da subdivisão.

A melhor forma de refinamento apontada na literatura é através da inserção de um novo ponto nodal na maior aresta, pois os elementos resultantes são de melhor qualidade e a degeneração dos tetraedros é menor[Golias94]. Como já dissemos, todos os elementos que tenham a aresta a ser dividida em comum devem ser subdivididos também, mantendo assim a conformidade da malha, conforme Figura 4.4.

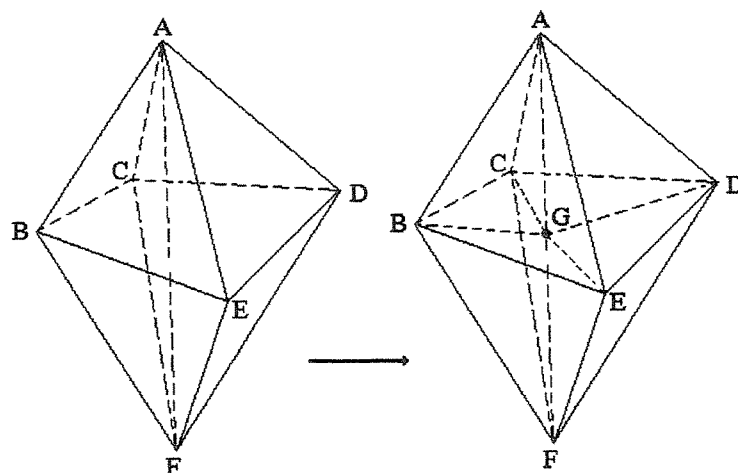


Figura 4.4 – Subdivisão de 4 tetraedros em 8 tetraedros, através da inserção de um novo nó G na aresta AF .

Da mesma forma que em nosso algoritmo bidimensional, esta inserção é feita na aresta de maior comprimento do tetraedro. Podemos utilizar os conceitos a respeito do refinamento de superfícies, descritos no Capítulo 3, para fazermos a inserção de nós em arestas situadas em superfícies curvas, tanto internas quanto externas. Nestes casos devemos tomar cuidado com o surgimento de elementos anômalos que podem comprometer a qualidade da malha e do nosso resultado numérico.

Da mesma maneira que ocorria no refinamento bidimensional, é necessário ter bastante cuidado para que não ocorram casos de elementos não conformes em nossa malha volumétrica. Na Figura 4.5 temos dois exemplos de não conformidades que podem ocorrer, caso aconteça uma falha na construção ou execução do programa.

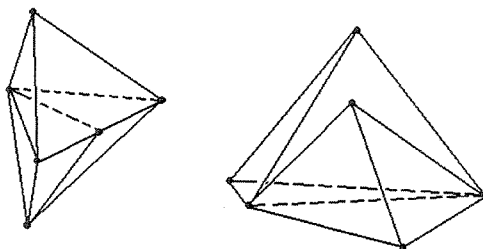


Figura 4.5 – Elementos tridimensionais não conformes.

4.2 – Transformações Topológicas

Na técnica bidimensional, para se saber se determinado elemento tinha um formato aceitável ou se era possível a troca de diagonais de Delaunay, era feito o teste do círculo vazio. Da mesma forma, é possível fazermos o teste em três dimensões, porém desta vez utilizando-se uma esfera, que é definida pelos quatro nós do tetraedro. É necessária, ainda, uma atualização da técnica de troca de diagonais de Delaunay, pois a mesma tende a não se sustentar em três dimensões, exceto em alguns casos específicos.

O principal problema que pode ocorrer na troca de diagonais envolvendo tetraedros é a alteração do contorno externo do poliedro formado pelos vértices, e do volume contido por eles. Isto acontece principalmente quando temos um poliedro côncavo. Um exemplo disto é mostrado na Figura 4.6, aonde a troca da diagonal BE pela diagonal AD , ou vice-versa, causa uma mudança drástica no contorno do poliedro. Como podemos ver, as faces ABE e BDE estão situadas em uma fronteira e a troca de diagonais entre os elementos $ABCE$, $BCDE$ e $ACDE$, $ABCD$ causa uma destruição do poliedro $ABCDE$ original. A transformação inversa também não é recomendada. Um parâmetro confiável para sabermos se uma transformação pode ser feita, é o cálculo do volume total do conjunto de tetraedros originais e do conjunto formado após a mudança. Se houver qualquer alteração no volume, isto significa que ocorreu uma alteração na forma do poliedro, e, portanto, a transformação não é possível.

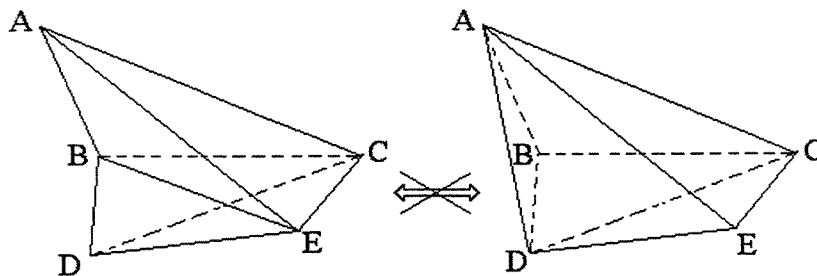


Figura 4.6 – Exemplos de trocas de diagonais impróprias em tetraedros.

Algumas transformações topológicas têm sido estudadas, entre elas podemos destacar a troca de diagonais externa e interna, a transformação de 2 elementos em 3, ou também chamada de transformação $2 \rightarrow 3$, e sua transformação inversa $3 \rightarrow 2$. Entretanto, estes métodos só podem ser utilizados se o conjunto de tetraedros a ser alterado formar um poliedro côncavo.

A troca de diagonais externas é, basicamente, uma extensão do caso 2D de troca de diagonais, e é aplicada quando a aresta a ser trocada está na superfície da estrutura. Para fazermos a troca basta considerarmos as faces que estão sobre a superfície e aplicar o teste da circunferência sobre eles. Tal procedimento é exemplificado na Figura 4.7. Na figura

inicial, à esquerda, temos os tetraedros $ABCD$ e $ABDE$. Após a troca, à direita, ficamos com os elementos $ABCE$ e $ACDE$. Este procedimento só é utilizado quando as faces formadas pelos pontos B , C , D e E estiverem localizadas na superfície da estrutura. Da mesma forma que no refinamento tridimensional superficial, é preciso cuidado para que esta troca de diagonais não provoque uma destruição do contorno da estrutura.

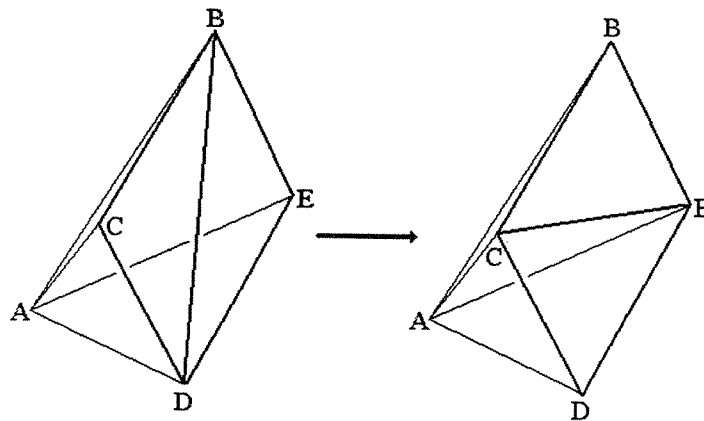


Figura 4.7 - Troca de diagonais externa.

A troca de diagonais interna só é possível nos casos em que temos quatro elementos que compartilham a mesma aresta e estão situados ao redor dela, de uma forma tal que ela esteja no centro do poliedro. Um exemplo bem simples pode ser visto na Figura 4.8(a). Nesta figura, todos os elementos estão dispostos ao redor da aresta AF . Após verificarmos que a troca de diagonais melhora a qualidade dos tetraedros e mantém o volume do poliedro inalterado, podemos efetuar a troca da antiga diagonal AF pela nova aresta CE (ou BD , qual for a aresta mais apropriada), obtendo a nova tetraedralização mostrada na Figura 4.8(b). Isto também costuma ser chamado de transformação $4 \rightarrow 4$.

No caso da transformação de dois tetraedros em três, ou $2 \rightarrow 3$, o processo começa com a identificação de dois elementos que possuem uma face comum entre si e que não satisfazem a propriedade da esfera. A geração de três elementos é feita através da eliminação desta face compartilhada pelos dois tetraedros e pela inserção de uma nova aresta que será compartilhada pelos três elementos. Deve-se tomar cuidado para que o poliedro formado pelos 2 elementos seja côncavo e para que esta aresta passe pelo interior

deste poliedro. Esta atenção é necessária para se prevenir a formação de elementos do tipo ‘lasca’. Na Figura 4.9 apresentamos um exemplo desta transformação. Nela a face BCD , Figura 4.9(a), é eliminada para a introdução da aresta AE , que é compartilhada pelos três elementos, Figura 4.9(b). Podemos ver que a aresta AE passa pelo interior do plano BCD .

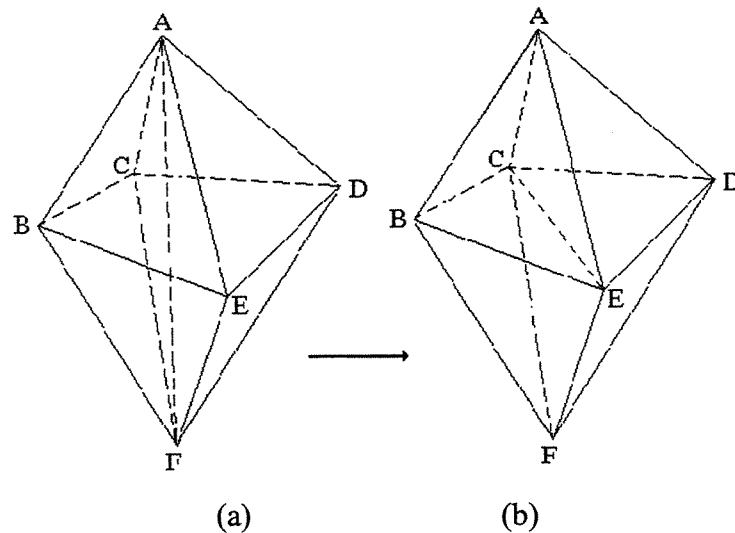


Figura 4.8 – Troca de diagonais interna em um conjunto de 4 tetraedros.

Este procedimento pode ser utilizado de forma reversa, quando, a partir de três elementos, geramos dois. Neste caso, o processo consiste em identificar os três elementos que compartilham determinada aresta, removê-la e introduzir uma face separando os dois novos elementos. Tomando como base a Figura 4.9(b), a aresta AE será eliminada e introduziremos a face BCD , Figura 4.9(a). Da mesma forma que na transformação anterior, deve-se tomar cuidado para que a aresta AE passe pelo interior da face BCD .

Outros tipos de transformações são descritos na literatura [Kettunen95], como, por exemplo, a transformação $5 \rightarrow 6$, na qual temos cinco elementos formando um poliedro côncavo e que compartilham uma aresta, Figura 4.10(a). Neste exemplo, todos os elementos estão ao redor da aresta AG , e os vértices restantes formam o plano $BCDEF$. Neste caso, existem cinco maneiras de se formarem as novas conexões, que são apresentadas diretamente sobre o plano $BCDEF$ na Figura 4.10(b), sendo necessário apenas

escolhermos aquela configuração que garanta a melhor qualidade para o conjunto de elementos assim formados. Usando processos similares, é possível fazermos outras modificações: $6 \rightarrow 8$, $7 \rightarrow 10$ e $8 \rightarrow 12$, embora em testes práticos algumas destas modificações demandem muito tempo de computação e a qualidade obtida não é muito melhor que a tida anteriormente [Kettunen95].

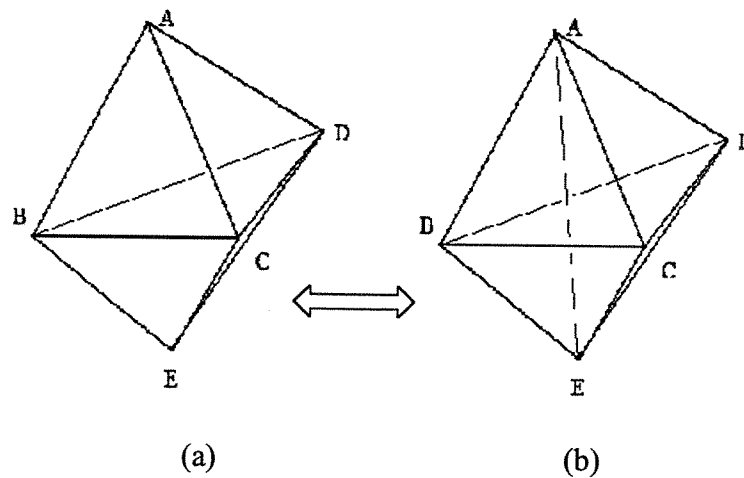


Figura 4.9 – Transformação topológica na qual fazemos a transformação de dois elementos em três e vice versa.

Estas transformações também podem ser obtidas através do uso conveniente das técnicas de transformação $2 \rightarrow 3$ e $3 \rightarrow 2$. Tomando uma transformação $4 \rightarrow 4$ como exemplo, vemos que são necessários 2 passos: primeiramente aplica-se a transformação $2 \rightarrow 3$ a um par de elementos, por exemplo, na Figura 4.8, transformamos os tetraedros $ABEF$ e $ABCF$ nos tetraedros $ABCE$, $BCEF$ e $ACEF$ através da exclusão da face ABF e da inclusão da aresta CE . A seguir, aplicamos uma transformação $3 \rightarrow 2$ no par restante de elementos, $ACDF$ e $ADEF$, e também ao tetraedro $ACEF$, obtendo o par de elementos $ACDE$ e $CDEF$, chegando assim ao mesmo resultado anterior. Observe-se que, neste processo, foi criado um elemento transitório $ACEF$ que possui volume igual a zero (criado na primeira transformação e eliminado na segunda), o que pode se tornar indesejável.

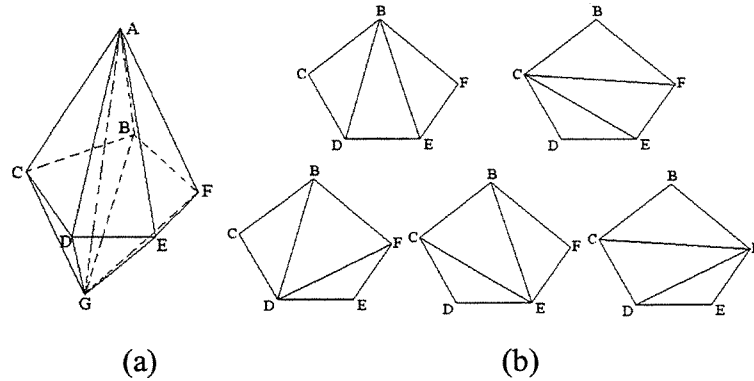


Figura 4.10 – Poliedro formado por cinco elementos ao redor da aresta AG e as cinco configurações possíveis para transformação.

Trabalhos mais recentes descrevem um operador de troca generalizado, que consiste na re-tetraedralização completa do poliedro formado pelos tetraedros que compartilham uma dada aresta, até achar a melhor configuração possível. Este processo é dispendioso em matéria de tempo de processamento, mas os resultados são muito eficientes para os propósitos de otimização de malhas [George99].

Outra forma de otimizar a qualidade geral da malha é fazendo a relaxação, ou suavização, da malha através do reposicionamento dos nós, conforme vimos na Seção 2.3.1. O método Laplaciano é o método mais utilizado, e consiste na determinação da nova posição do nó através da média das coordenadas dos nós conectados a ele. Por conveniência, repetiremos aqui a fórmula para o cálculo da nova posição do nó.

$$P^* = (1/n) \sum_{i=1,2,\dots,n} P_i \quad (2.13)$$

onde P^* é a nova coordenada do nó sendo considerado, e P_i são as posições dos n nós conectados a P^* . Na Figura 4.11 vemos uma ilustração do processo de suavização da posição dos nós usando elementos tetraédricos. A literatura nos indica que a utilização desta técnica, em casos tridimensionais volumétricos, requer mais atenção que no caso bidimensional, pois é necessário conferir se a movimentação do nó realmente aumenta a qualidade dos elementos, e se a nova posição do nó continua dentro do poliedro formado

pelos elementos, o que pode não ocorrer quando tratamos de poliedros não-convexos [Golias94].

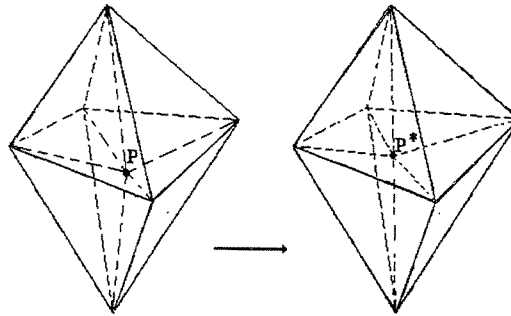


Figura 4.11 – Relaxação da posição dos nós em três dimensões.

Outra diferença em relação ao caso bidimensional é que agora temos quatro tipos de nós, de acordo com a sua posição na malha tetraédrica. Eles são classificados em: nós internos, que ficam no interior do domínio do problema; nós de superfície, que caem nas interfaces internas ou externas do domínio; nós de aresta, que ficam nas junções entre duas superfícies, e, finalmente, nós de canto, que são aqueles que estão na junção entre três ou mais superfícies. Na Figura 4.12, vemos um exemplo desta classificação. Para evitar a destruição das superfícies durante o processo de relaxação, movemos apenas os nós internos da estrutura.

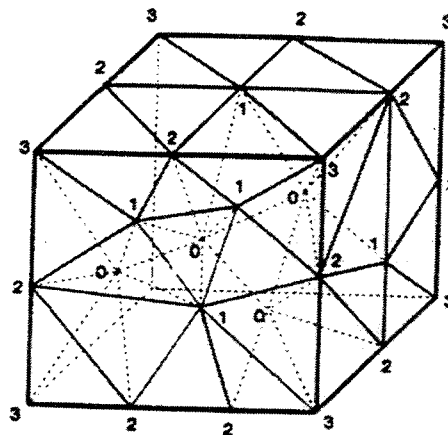


Figura 4.12 – Classificação dos nós de acordo com sua posição na malha: 0 – internos, 1 – superficiais, 2 – de aresta e 3 – de canto.

4.3 – Implementação do Algoritmo

Com algumas mudanças, foi possível adaptar o nosso algoritmo de refinamento tridimensional superficial para o refinamento de estruturas tridimensionais volumétricas. Estas mudanças ocorreram principalmente na estrutura de dados, na rotina de divisão dos elementos e no processo de melhoria da qualidade da malha, com a implementação de algoritmos de trocas de diagonais externa e de relaxamento da malha.

A mudança na estrutura de dados reflete o fato de que à medida que aumentamos a complexidade de nossas estruturas, necessitamos de mais informações para a correta descrição das mesmas, condição essencial para que o refinamento ocorra sem problemas. Como vimos, no caso bidimensional é preciso armazenar dados a respeito dos elementos, ou seja, os nós que formam o triângulo, as coordenadas cartesianas dos nós e os elementos vizinhos a cada triângulo. Já na estrutura tridimensional superficial são necessários todos estes dados, e ainda informações a respeito das superfícies que compõem a estrutura em questão.

Para o caso tridimensional volumétrico, a quantidade de dados manipulados cresce ainda mais, pois é necessário mantermos registros de todas as arestas pertencentes a um elemento, de quais elementos tem determinada aresta em comum e, também, dados a respeito da posição dos nós na malha, para efeito de otimização. Tais registros aumentam a quantidade de memória ocupada do computador, mas são extremamente úteis, pois diminuem o tempo de procura de determinadas rotinas, diminuindo assim o tempo de computação. Maiores detalhes a respeito da estrutura de dados utilizada podem ser encontrados no Apêndice A.

De maneira análoga ao caso bidimensional e tridimensional superficial, o algoritmo se inicia com a criação de uma lista contendo todos os elementos que precisam ser subdivididos. Iniciando com primeiro elemento da lista, faz-se a subdivisão de todos os tetraedros pertencentes a ela e dos elementos resultantes da subdivisão que estejam acima

da condição limite, até que a lista esteja vazia. Após chegarmos ao final da lista de subdivisão, fazemos a estabilização prévia da malha, que consiste na aplicação das transformações topológicas e de relaxação dos nós certo número de vezes. Se o critério de parada ainda não foi atingido, retomamos todo o processo. Após o final do refinamento, aplicamos uma estabilização final da malha com a repetida aplicação das transformações topológicas e de relaxação dos nós até que não ocorram mais mudanças significativas na malha [Golias94]. Na Figura 4.13, vemos um fluxograma do algoritmo, com suas fases distintas. As etapas de refinamento e estabilização são comentadas mais detalhadamente a seguir.

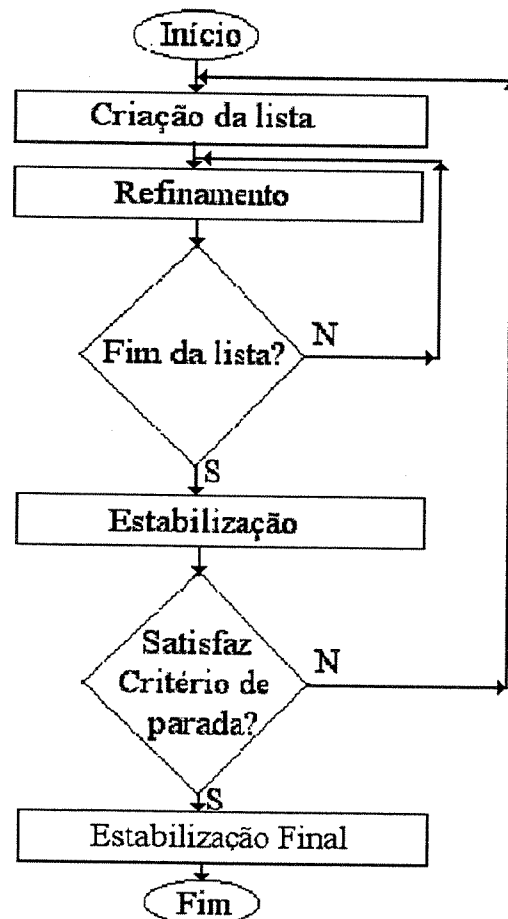


Figura 4.13 – Fluxograma da implementação do programa de refinamento 3D volumétrico.

A rotina de subdivisão é o ponto principal do algoritmo, e é baseada no método utilizado nos algoritmos de refinamento superficial 2D e 3D. Novamente, neste caso nos

valeremos da terminologia introduzida por Rivara para fazermos algumas definições [Rivara97b]. De maneira análoga ao **LEPP**, é necessário se achar o tetraedro com maior aresta em determinada região. Assim, partindo de um determinado tetraedro t_0 , encontramos a sua aresta de maior comprimento A_0 e, então, fazemos uma varredura em todos os elementos que compartilham esta aresta para determinar se algum deles possui uma aresta maior que A_0 . Se a aresta de t_0 for a maior, então podemos fazer a divisão deste elemento e dos elementos que a compartilham, através da inserção de um nó no ponto médio da aresta A_0 , conforme vimos na Seção 4.1. Caso o elemento t_0 não possua a maior aresta do conjunto, que pertence a um dado elemento t_1 , então colocamos o elemento t_1 no topo da lista para subdivisão e fazemos uma nova checagem com os elementos que compartilham a aresta A_1 entre si. O processo é repetido até se chegar a um determinado elemento t_n que possua uma aresta A_n que é a maior aresta nesta região. Os elementos que compartilham esta aresta entre si são denominados de conjunto de tetraedros terminais.

Ao fazermos a subdivisão de cada grupo de elementos, é necessário retirar da relação de elementos para partição aqueles elementos pertencentes à lista e que foram divididos juntamente com t_n . Fazemos isto a fim de evitar a divisão desnecessária de um elemento mais de uma vez, o que pode implicar em elementos de aspecto pobre. Após a subdivisão, aplicamos as transformações topológicas a todos elementos subdivididos e resultantes, e finalmente checamos se estão acima da condição limite, caso em que os adicionamos à lista para nova subdivisão. O fluxograma do processo de refinamento é apresentado na Figura 4.14.

Graças à estrutura de dados utilizada por nós, a procura pelos elementos que possuam a maior aresta é feita de maneira rápida e eficiente. Com o objetivo de se reduzir o tempo de computação no processo de busca, uma limitação foi imposta: quando a diferença entre a maior aresta e a aresta atualmente sendo testada for³ pequena, digamos 1%, então nenhuma troca é feita, assumindo-se que a aresta atual é a maior do conjunto. Isto implica em um pequeno erro, que é corrigido através da aplicação das transformações topológicas aos elementos da malha.

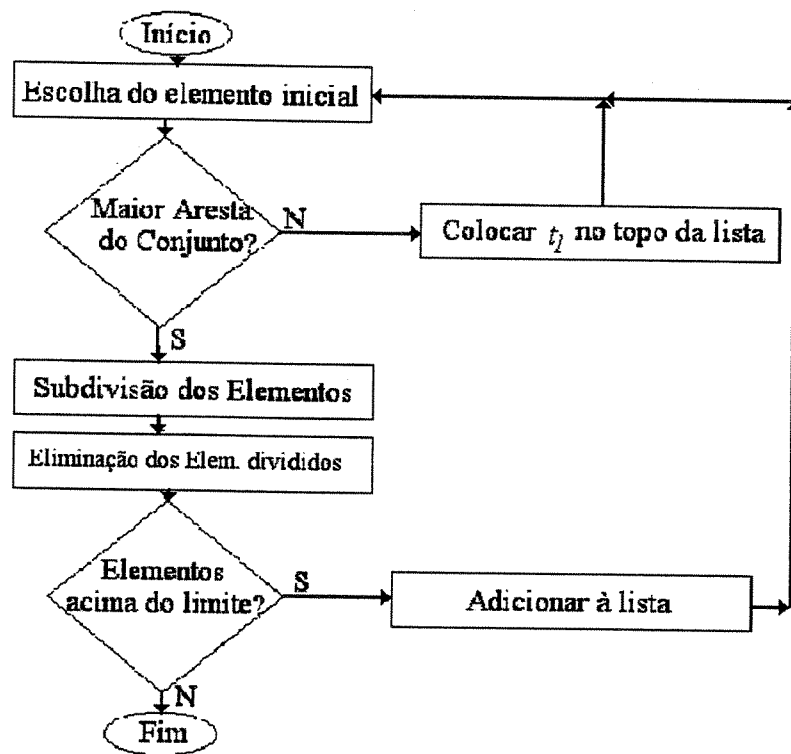


Figura 4.14 – Fluxograma da subrotina de refinamento.

O processo todo é repetido até se chegar ao critério de parada escolhido, que pode ser qualquer um daqueles citados na Seção 2.2. Então, como foi dito acima, fazemos uma estabilização final da malha.

O processo de estabilização consiste na aplicação das transformações topológicas, descritas na Seção 4.3, seguido da suavização da posição dos nós da malha. Isto é repetido certo número de vezes, ou até que nenhuma modificação importante seja feita, isto é, até que a malha se estabilize. Este processo é considerado a chave para a produção de malhas de boa qualidade, pois primeiramente as transformações topológicas tornam a malha concordante com as propriedades de Delaunay, e depois a relaxação reposiciona os nós e embora a qualidade geral da malha seja melhorada, a malha é perturbada e em alguns casos ela não atende mais às condições de Delaunay. Isto leva à repetição do processo, até que se obtenha a convergência, que ocorre quando acontecem poucas transformações e os nós se movem distâncias pequenas. Geralmente esta estabilização ocorre após repetirmos entre 5 e

10 vezes o processo. Na Figura 4.15, apresentamos um fluxograma do processo de relaxação.

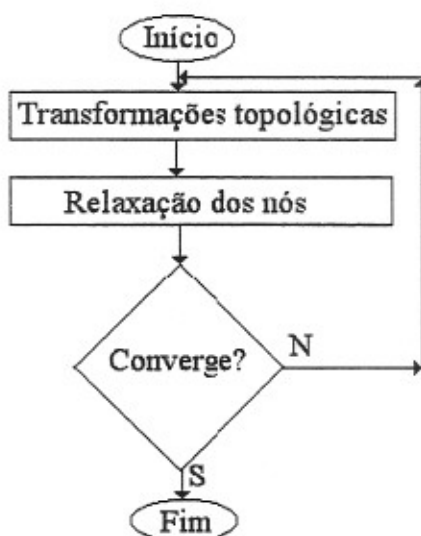


Figura 4.15 – Fluxograma do processo de Relaxação.

Alguns exemplos de malhas obtidas com nosso algoritmo podem ser vistos na Seção 6.4. Também foi feito um teste prático utilizando uma malha obtida com nosso programa.

Capítulo 5

Métodos Numéricos

Este Capítulo traz informações a respeito de alguns métodos utilizados na análise eletromagnética. Existem diferentes técnicas de modelagem numérica, cada uma com muitas variantes para o estudo de casos específicos. Entre elas podemos citar o método dos elementos finitos, o método das diferenças finitas, etc. Nesta tese nós nos concentraremos em três técnicas diferentes que foram utilizadas para a obtenção dos resultados numéricos que serão mostrados no Capítulo 6: O Método dos Elementos Finitos, o Método dos Elementos de Aresta e o Método dos Momentos.

Primeiramente, falaremos a respeito do Método de Elementos Finitos Nodais, que, certamente, é o mais conhecido e utilizado atualmente nas simulações eletromagnéticas. Também mostraremos um caso específico desta técnica, adaptado para a simulação de guias de onda não-lineares. Os próximos métodos que estudaremos são o Método dos Momentos,

adaptado para a simulação do espalhamento de uma onda eletromagnética, e o Método dos Elementos de Aresta.

Nosso objetivo é apresentar de forma abreviada a formulação matemática, sem uma demonstração detalhada, que pode ser amplamente encontrada na literatura [Bastos92, Jin93, Koshiba92, Silvester96, Wang91]. Nos ateremos somente às demonstrações básicas para um melhor entendimento do problema físico a ser resolvido, das necessidades do método utilizado e de sua implementação.

5.1 – As Equações de Maxwell e a Equação de Onda

O modelamento de estruturas para análise eletromagnética envolve, tipicamente, a solução das Equações de Maxwell, sobre um domínio finito [Lee86]. As equações de Maxwell, em sua forma diferencial, para um meio isotrópico livre de cargas e correntes são dadas por:

$$\nabla \cdot \mathbf{D} = 0 \quad (5.1a)$$

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \quad (5.1b)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (5.1c)$$

$$\nabla \times \mathbf{H} = \varepsilon \frac{\partial \mathbf{E}}{\partial t} \quad (5.1d)$$

com $\mathbf{H}$ e $\mathbf{E}$ sendo as intensidades de campo magnético e elétrico, respectivamente [Jin93], e os termos ε e μ sendo a permissividade elétrica e a permeabilidade magnética do meio.

Aplicando-se a condição de guiamento fraco, o acoplamento entre os componentes dos campos podem ser ignorados, produzindo as equações de Helmholtz, ou equações de movimento de onda, abaixo:

$$\nabla^2 \mathbf{E} = \mu\epsilon \frac{\partial^2 \mathbf{E}}{\partial t^2} \quad \text{e} \quad \nabla^2 \mathbf{H} = \mu\epsilon \frac{\partial^2 \mathbf{H}}{\partial t^2} \quad (5.2)$$

Sendo equações vetoriais, elas equivalem a seis equações escalares, uma para cada componente. Existem diversas soluções para estas equações, correspondendo a diferentes tipos de ondas eletromagnéticas – planas, esféricas, etc. . Consideremos aqui dois casos bastante simples de ondas planas monocromáticas que se deslocam na direção longitudinal z , que podem ser descritas pelas equações:

$$\mathbf{E}(x, y, z) = \mathbf{E}(x, y) e^{j(\omega t - \beta k_0 z)} \quad (5.3a)$$

$$\mathbf{H}(x, y, z) = \mathbf{H}(x, y) e^{j(\omega t - \beta k_0 z)} \quad (5.3b)$$

onde ω é a frequência, β é o índice efetivo do meio, k_0 é a constante de propagação da onda no vácuo e j é o número imaginário correspondente a $\sqrt{-1}$. Após tratarmos cada um dos diversos componentes vetoriais, fazendo algumas substituições, e usando o fato de que $\omega = k_0 c$ (c sendo a velocidade da luz no vácuo) chegamos às equações escalares finais que são denominadas soluções, ou modos, transversal elétrica (TE) e transversal magnética (TM):

$$TE: \frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2} + (n^2 - \beta^2) k_0 E = 0 \quad (5.4a)$$

$$TM: \frac{\partial^2 H}{\partial x^2} + \frac{\partial^2 H}{\partial y^2} + (n^2 - \beta^2) k_0 H = 0 \quad (5.4b)$$

onde n é o índice de refração do material, e é dado por $n^2 = (\epsilon\mu)/(\epsilon_0 \mu_0)$. Estas equações são chamadas de transversais pois utilizamos as componentes transversais dos campos em

consideração, elétrico ou magnético, e que estão totalmente contidos no plano transversal à direção de propagação.

5.2 – Método dos Elementos Finitos

O princípio do método de elementos finitos consiste na substituição de um domínio contínuo por um conjunto de subdomínios discreto, nos quais a nossa função incógnita é representada por uma interpolação de funções com coeficientes desconhecidos. Desta forma, nosso problema de valores de contorno original, que tinha um número infinito de graus de liberdade, é transformado em um sistema com um número finito de incógnitas [Jin93, Rahman91]. Com a aplicação de um procedimento variacional de Ritz ou Galerkin, montamos um sistema de equações algébricas que, após resolvidas, nos fornece a solução para o problema de valores de contorno. Resumidamente, podemos decompor a resolução de um problema de valores de contorno em 4 etapas:

- 1 – Discretização do domínio.
- 2 – Seleção das funções de interpolação.
- 3 – Formulação do sistema de equações.
- 4 – Solução do sistema de equações.

O primeiro passo já foi discutido amplamente nos Capítulos 2, 3 e 4. A seguir, mostraremos uma apresentação bastante sucinta do método de elementos finitos unidimensional. Embora seja uma apresentação simples, ela serve para demonstrar o funcionamento dos principais conceitos envolvidos, e pode ser estendida para duas e três dimensões, para elementos triangulares, retangulares, tetraédricos, etc. Uma descrição mais profunda pode facilmente ser encontrada na literatura, conforme referências [Bastos92, Jin93, Koshiba92, Silvester96].

As funções de interpolação podem assumir várias formas. A mais utilizada é a função de primeira ordem aonde consideramos que o campo elétrico E^e varia linearmente no interior do e -ésimo elemento, seguindo a equação

$$E^e(x) = a^e + b^e x \quad (5.5)$$

aonde a^e e b^e são as constantes a serem determinadas. Existem funções de interpolação mais complexas, aonde o campo pode variar de forma quadrática, ou mesmo cúbica. Tecnicamente, os elementos finitos mais sofisticados podem produzir melhores resultados, mas a correta aplicação dos elementos lineares produz excelentes resultados e viabiliza a elaboração de sistemas de cálculo eficazes sem que seja necessário um processo de desenvolvimento excessivamente penoso.

Unidimensionalmente, a nossa malha terá a forma de uma reta composta de vários segmentos, que são os elementos, conforme Figura 5.1. Cada elemento será formado por dois nós e um segmento de reta que os conecta.

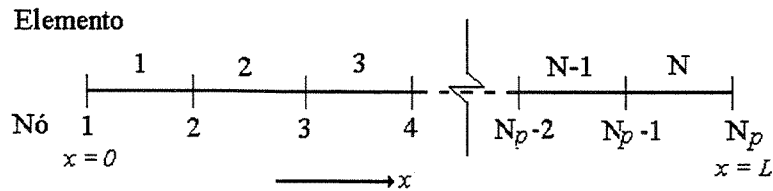


Figura 5.1 – Domínio unidimensional de comprimento L dividido em N elementos, e com N_p pontos nodais.

Analisando um elemento e qualquer isoladamente, cujos pontos nodais P_1 e P_2 são localmente numerados como nós 1 e 2, veremos que as equações para o campo elétrico serão

$$E_1^e = a^e + b^e x_1^e \quad (5.6a)$$

$$E_2^e = a^e + b^e x_2^e \quad (5.6b)$$

onde E_1^e e E_2^e são os valores do campo elétrico nos pontos 1 e 2, respectivamente.

Resolvendo este pequeno sistema para encontrarmos o valor das constantes a^e e b^e , e substituindo nas Equações 5.6, nós obtemos

$$E^e(x) = \sum_{j=1}^2 N_j^e(x) E_j^e \quad (5.7)$$

na qual N_j^e denotam as funções de base, ou de interpolação, lineares dadas por

$$N_1^e = \frac{x_2^e - x}{x_2^e - x_1^e} \quad \text{e} \quad N_2^e = \frac{x - x_1^e}{x_2^e - x_1^e} \quad (5.8)$$

Estas funções assumem a forma de um delta de Kronecker, isto é, podemos escrevê-las como $N_j^e(x_i^e) = \delta_{ij}$, com $\delta_{ij} = 1$ para $i = j$, e $\delta_{ij} = 0$ para $i \neq j$, conforme vemos nas Figuras 5.2.

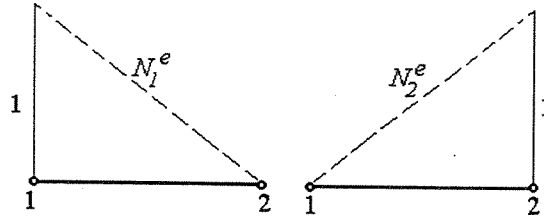


Figura 5.2 – Funções de interpolação linear unidimensional.

O próximo passo é a formulação do sistema de equações que pode ser feito através de dois métodos: Ritz ou Galerkin. Para poupar o leitor de uma demonstração complicada, que pode facilmente ser encontrada na literatura, falaremos apenas que a aplicação das formulações de Ritz ou Galerkin nos leva a um sistema de equações para os campos incógnitos E_i ($i = 1, 2, \dots, N$).

O passo final é a resolução do sistema de equações obtido. Neste caso, nós obtemos um sistema de equações que produz uma matriz do tipo tridiagonal, isto é, os únicos elementos não iguais a zero são aqueles pertencentes à diagonal principal e às diagonais

imediatamente superior e inferior, isto é, os elementos da matriz M cujos índices correspondam a m_{ii} , $m_{i,i-1}$ e $m_{i,i+1}$ (que é igual a $m_{i+1,i}$). Estes sistemas são facilmente resolvidos através de diversas técnicas tais como eliminação Gaussiana e Iteração de Sub-espaço [Itoh96, Silvester96].

5.2.1 – Aplicações do Método de Elementos Finitos

Usando o método dos elementos finitos aplicado à formulação fraca de Galerkin, a Equação 5.4(a) é transformada em um problema de autovalores do tipo

$$Au = \lambda Bu \quad (5.9)$$

onde A e B são matrizes esparsas e B é simétrica. O autovalor λ , nesse caso, é igual a β^2 e os elementos de cada matriz são dados por:

$$(A)_{ij} = -\int_{\Omega} (\nabla \varphi_i \cdot \nabla \varphi_j + k_0^2 n^2 \varphi_i \cdot \varphi_j) dx dy \quad (5.10a)$$

$$(B)_{ij} = k_0^2 \int_{\Omega} \varphi_i \cdot \varphi_j dx dy \quad (5.10b)$$

nos quais φ_i e φ_j são as funções de base e Ω é o domínio truncado contendo a seção reta do guia. Considerou-se condições de contorno de Dirichlet (TE) e Newmann (TM) nulas.

Para o problema não-linear, a aproximação é semelhante, porém agora o índice de refração do material não linear será uma função da potência do campo no guia [Hayata88, Li92]. Assim, fazendo-se as substituições necessárias na Equação 5.4(a), chegamos a

$$\frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2} + (n^2(x, y) + f(\alpha E^2) - \beta^2) k_0^2 E = 0 \quad (5.11)$$

na qual a função $f(\alpha E^2)$ descreve a não-linearidade do guia. O coeficiente α é dado por $\alpha = n_3^2 n_K c \epsilon_0$, onde n_3 é o índice de refração do material não-linear para baixas potências, n_K é o coeficiente não-linear de Kerr do meio, c é a velocidade da luz no vácuo e ϵ_0 é a constante de permissividade elétrica no vácuo. Para os cálculos mostrados na Seção 6.2.3.2, supomos que o material não-linear tinha uma saturação de forma gaussiana, caracterizada por:

$$f(\alpha E^2) = 2n_3 \Delta n_{sat} \left[1 - \exp\left(-\frac{\alpha E^2}{2n_3 \Delta n_{sat}}\right) \right] \quad (5.11)$$

com Δn_{sat} sendo o máximo aumento não-linear. Para o caso da condição de guiamento fraco, isto nos dá um comportamento semelhante ao comportamento não-linear de Kerr. Supondo que o campo é dado por $E(x, y) = A\phi(x, y)$, onde ϕ é o campo normalizado que

satisfaz a equação $\int_{-\infty-\infty}^{\infty} \int_{-\infty-\infty}^{\infty} \phi^2(x, y) dx dy = 1$, então a potência modal é dada por:

$$P = \frac{\beta c \epsilon_0}{2k_0} A^2 \quad (5.12)$$

e a função ϕ satisfaz

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + (n^2(x, y) + f(\alpha A^2 \phi^2) - \beta^2) k_0^2 \phi = 0 \quad (5.13)$$

Utilizando a técnica proposta por Li e outros [Li92], o que fazemos é partindo de um dado valor para β , calculamos uma primeira aproximação para os valores de A e ϕ , utilizando o método dos elementos finitos. Como primeira aproximação dos campos, A_0 e ϕ_0 , utilizamos a solução do problema para o caso linear, no qual supomos que a função de não-linearidade é igual a zero. Usando estes valores como uma forma de realimentação do esquema iterativo, chegamos à Equação 5.14, na qual o autovalor é a magnitude de campo A_{m+1} .

$$\frac{\partial^2 \phi_{m+1}}{\partial x^2} + \frac{\partial^2 \phi_{m+1}}{\partial y^2} + \left[n^2(x, y) + A_{m+1}^2 \left(\frac{f(\alpha A_m^2 \phi_m^2)}{A_m^2} \right) - \beta^2 \right] k_0^2 \phi_{m+1} = 0 \quad (5.14)$$

Esta equação é resolvida iterativamente utilizando um programa de cálculo através de elementos finitos, até que a solução atinja a convergência e estabilize numericamente. Ao final de cada passo, o campo ϕ é normalizado. Assim, uma vez obtido o valor para a magnitude do campo elétrico dentro da aproximação desejada, basta calcular a potência modal através da Equação 5.12.

5.3 – Método dos Elementos Finitos de Aresta

Esta nova aproximação, também chamada de Método dos Elementos Vetoriais, começou a ser utilizada a partir da década de 80 para resolver alguns problemas que o método dos elementos finitos nodais não conseguia. Alguns exemplos claros da limitação do método nodal tradicional são a contaminação das soluções por modos espúrios sem correlação física, as singularidades de campo associadas às arestas e cantos, em meios condutores ou dielétricos, de um domínio e as condições de contorno internas em um meio não homogêneo. Felizmente, com o uso deste método tais problemas foram superados, e o uso da técnica tem aumentado consideravelmente. A única desvantagem do método é o aumento do número de incógnitas para uma dada malha.

Nesta seção, faremos uma breve introdução ao método, utilizando elementos triangulares bidimensionais. A extensão para o caso tridimensional pode ser encontrada na literatura referenciada [Jin93, Golias94b, Lee95].

Como vimos, o Método dos Elementos de Aresta também é chamado de Método dos Elementos Vetoriais, pois associa funções de base vetoriais às arestas, e não aos nós, como

era feito no método dos elementos finitos tradicional. Assim, no triângulo mostrado na Figura 5.3, cada aresta possuirá uma função de base independente.

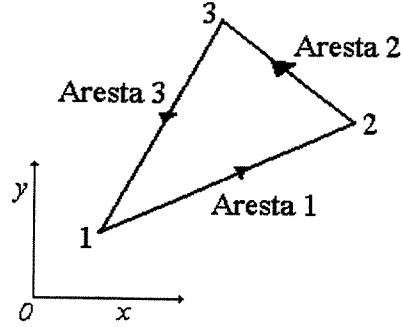


Figura 5.3 – Elemento triangular de aresta.

Se estivéssemos utilizando elementos retangulares, seria fácil assinalar os campos tangenciais E_x e E_y a cada uma das arestas, devido ao próprio formato do elemento, pois cada um dos lados do retângulo seria paralelo a um dos eixos coordenados x e y . No caso de elementos triangulares, tal já não é possível, pois ao menos uma das arestas não será paralela aos eixos x e y , sendo necessário derivar cada um dos componentes de acordo com suas propriedades.

Necessitaremos da definição das funções de interpolação linear do elemento triangular bidimensional da Figura 5.3, dadas por

$$L_j^e(x, y) = \frac{1}{2A^e} (a_j^e + b_j^e x + c_j^e y) \quad (5.15)$$

onde A_e é a área do elemento e , e os coeficientes a , b e c , para cada nó do elemento, são dados por

$$\begin{array}{lll} a_1^e = x_2^e y_3^e - x_3^e y_2^e & b_1^e = y_2^e - y_3^e & c_1^e = x_3^e - x_2^e \\ a_2^e = x_3^e y_1^e - x_1^e y_3^e & b_2^e = y_3^e - y_1^e & c_2^e = x_1^e - x_3^e \\ a_3^e = x_1^e y_2^e - x_2^e y_1^e & b_3^e = y_1^e - y_2^e & c_3^e = x_2^e - x_1^e \end{array}$$

Consideremos agora a função vetorial dada por

$$W_{12} = L_1^e \nabla L_2^e - L_2^e \nabla L_1^e \quad (5.16)$$

é possível demonstrar que $\nabla \cdot W_{12} = 0$, e $\nabla \times W_{12} = 2(\nabla L_1^e \times \nabla L_2^e)$.

Supondo que $\mathbf{e}_1$ é um vetor unitário cuja direção aponta do ponto nodal 1 para o nó 2, e sabendo que L_1^e é uma função linear que varia de 1 no nó 1 até zero no nó 2, e que L_2^e também é uma função linear e varia de 1 no ponto nodal 2 até zero em 1, nós temos que $\mathbf{e}_1 \cdot \nabla L_1^e = -1/\ell_1^e$ e $\mathbf{e}_1 \cdot \nabla L_2^e = 1/\ell_1^e$, onde ℓ_1^e é o comprimento da aresta conectando os pontos 1 e 2. A partir disto é possível mostrar que

$$\mathbf{e}_1 \cdot W_{12} = \frac{L_1^e + L_2^e}{\ell_1^e} = \frac{1}{\ell_1^e} \quad (5.17)$$

Esta expressão significa que W_{12} possui uma componente tangencial constante ao longo da aresta (1,2). Mais ainda, como L_1^e desaparece ao longo da aresta (2,3), e L_2^e desaparece ao longo da aresta (1,3), W_{12} não terá qualquer componente tangencial nestas duas arestas. Deste modo, W_{12} possui todos os requisitos necessários para ser uma base vetorial para o campo associado à aresta (1,2). Se nós definimos esta aresta como sendo a aresta 1, a aresta (2,3) como aresta 2 e a aresta (1,3) como aresta 3, nós ficamos com um conjunto de funções vetoriais de base, dadas por

$$N_1^e = W_{12} \ell_1^e = (L_1^e \nabla L_2^e - L_2^e \nabla L_1^e) \ell_1^e \quad (5.18a)$$

$$N_2^e = W_{23} \ell_2^e = (L_2^e \nabla L_3^e - L_3^e \nabla L_2^e) \ell_2^e \quad (5.18b)$$

$$N_3^e = W_{13} \ell_3^e = (L_3^e \nabla L_1^e - L_1^e \nabla L_3^e) \ell_3^e \quad (5.18c)$$

onde ℓ_j^e é incluído para normalizar cada função N_j^e e torna-la adimensional. Escrevendo estas equações de uma forma mais compacta, então o campo elétrico vetorial dentro do elemento ficará

$$E^e = \sum_{i=1}^3 N_i^e E_i^e \quad (5.19)$$

com o campo na i -ésima aresta sendo denotado por E_i^e .

Na Figura 5.4, nós mostramos os vetores destas funções sobrepostas a um elemento triangular típico. Da forma esperada, estas funções vetoriais de base possuem componentes tangenciais somente nas arestas associadas. Estas componentes são contínuas entre os elementos, ao passo que admite-se que a componente normal pode variar abruptamente de um elemento para outro, de forma similar aos campos eletromagnéticos existentes nas interfaces entre dois meios diferentes, por exemplo.

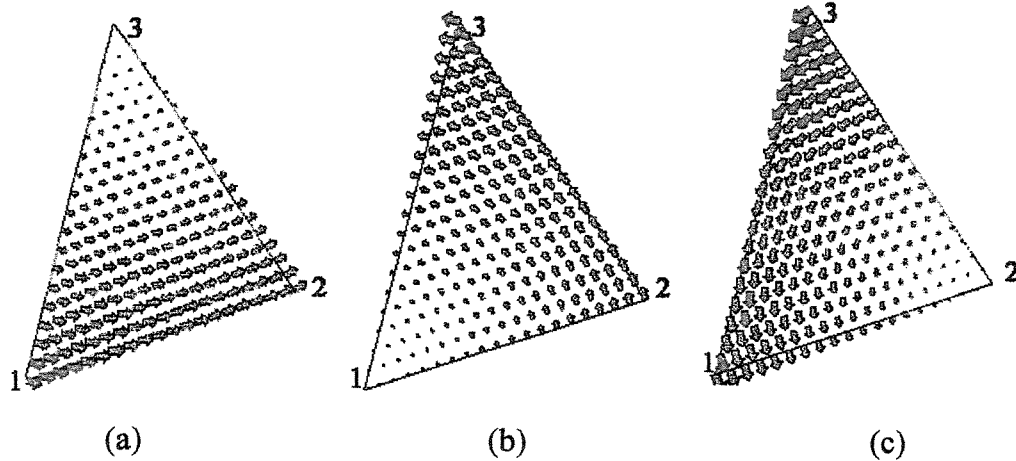


Figura 5.4 – Funções de base vetoriais associadas às arestas de um elemento triangular: (a) N_1^e , (b) N_2^e e (c) N_3^e [Bomholt94].

De forma semelhante ao empregado no método dos elementos finitos, a aplicação dos métodos de Ritz ou Galerkin, nos proporcionará um conjunto de equações que, após resolvidas, nos proporcionarão a resposta ao problema em estudo.

5.3 – Método dos Momentos

Para estudarmos a validade das malhas tridimensionais superficiais, utilizamos um programa que calcula o espalhamento de ondas eletromagnéticas por um corpo condutor tridimensional, que pode ser de formato arbitrário.

Para isto é utilizado o Método dos Momentos [Harrington68, Wang91]. A resolução baseia-se na divisão da superfície em pequenas regiões, preferencialmente triangulares devido à maior adaptabilidade a superfícies arbitrárias, pois tratam-se das menores superfícies bidimensionais, e portanto mais fáceis de se adaptarem a superfícies quaisquer. A corrente superficial em cada região é calculada usando uma das Equações Integrais de Superfície, que podem ser divididas em duas classes: ‘*Electric Field Integral Equation*’, *EFIE* (Equação Integral de Campo Elétrico), Equação 5.20a, e ‘*Magnetic Field Integral Equation*’, *MFIE* (Equação Integral de Campo Magnético), Equação 5.20b. Ambas são aplicadas ao estudo de campos em estruturas tridimensionais de formato arbitrário. A diferença entre ambas é que a equação magnética é aplicada apenas em estruturas fechadas e suaves, enquanto que a equação elétrica também pode ser aplicada a estruturas abertas.

$$\hat{n} \times \mathbf{E}^i(\mathbf{r}) = \hat{n} \times \int_S \left[j\omega\mu\mathbf{J}_s g + \frac{1}{j\omega\epsilon} (\nabla'_s \cdot \mathbf{J}_s) \nabla'_s g \right] ds' \quad (5.20a)$$

$$\mathbf{J}_s = 2\hat{n} \times \mathbf{H}^i(\mathbf{r}) + 2\hat{n} \times \int_S \mathbf{J}_s \times \nabla'_s g ds' \quad (5.20b)$$

Nestas equações, $\hat{n}$ é um vetor unitário normal à superfície S da estrutura, $\mathbf{E}^i$ e $\mathbf{H}^i$ são os campos elétrico e magnético da onda incidente, ω é a sua frequência angular, $\mathbf{J}_s$ é a corrente superficial no objeto e g é a função escalar de Green, dada por:

$$g(\mathbf{r}, \mathbf{r}') = \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \quad (5.21)$$

No nosso caso, estudaremos a incidência de uma onda eletromagnética plana em uma esfera condutora. Como a esfera é uma superfície simples, suave e fechada, utilizaremos a Equação Integral de Campo Magnético, pois a mesma produz bons resultados e a modelagem necessária é mais simples.

Começaremos com a aproximação da corrente superficial $\mathbf{J}_s$ como sendo a soma das correntes de cada uma das sub-regiões, ou seja:

$$\mathbf{J}_s(\mathbf{r}) = \sum_{n=1}^N \mathbf{J}_n(\mathbf{r}) \quad (5.22)$$

onde $\mathbf{J}_n(\mathbf{r}) = I_n^1 \mathbf{P}_n^1(\mathbf{r}) + I_n^2 \mathbf{P}_n^2(\mathbf{r})$ e I é a corrente no elemento e

$$\mathbf{P}_n^i(\mathbf{r}) = \begin{cases} \hat{u}_n^i & \text{para } \mathbf{r} \in \Delta S_n \\ 0 & \text{outro lugar} \end{cases} \quad \text{para } i=1, 2; \quad n=1, 2, \dots, N.$$

o termo ΔS_n é o n -ésimo triângulo no qual a corrente deve ser determinada. Os vetores unitários ortonormais $\hat{u}_n^i$ são escolhidos arbitrariamente e estão situados no mesmo plano de ΔS_n .

Na Figura 5.5 vemos uma pequena ilustração do nosso problema. Na figura mostramos apenas 2 elementos triangulares para melhor visualização. Podemos ver ainda a onda eletromagnética incidente e o vetor $\hat{n}$ normal à superfície.

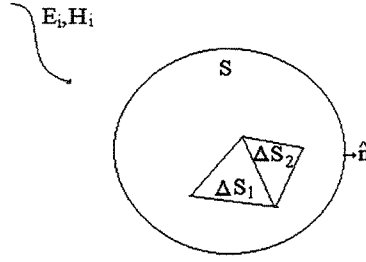


Figura 5.5 – Figura ilustrando a superfície S , e alguns componentes do problema.

Após uma série de cálculos, que podem ser acompanhados na referência [Wang91], chegamos ao conjunto final de equações a serem resolvidas:

$$\sum_{n=1}^N \sum_{j=1}^2 I_n^j Z_{mn}^{ij} = V_m^i \quad i = 1, 2; \quad m = 1, 2, \dots, N \quad (5.23)$$

onde

$$V_m^i = \hat{u}_m^i \cdot 2\hat{n}_m \times \mathbf{H}^i(\mathbf{r}_m)$$

$$Z_{mn}^{ij} = \delta_m^n \delta_i^j - \frac{1 + jkR_{mn}}{2\pi R_{mn}} e^{-jkR_{mn}} \Delta S_n \left[\hat{u}_m^i \cdot \hat{n}_m \times (\hat{u}_n^j \times \hat{R}_{mn}) \right] (1 - \delta_m^n)$$

Nestas equações, R_{mn} é a distância entre os triângulos m e n , $\hat{R}_{mn}$ é o seu versor unitário e δ_m^n é o delta de Kronecker, sendo igual a 1 quando $m = n$ e zero em outro caso.

Para avaliarmos V_m^i , supomos que se $\mathbf{H}^i$ for uma onda plana com uma direção de propagação em coordenadas esféricas (θ_0, ϕ_0) e uma constante de propagação vetorial $\mathbf{k}$, como vemos na Figura 5.6, nós podemos definir

$$\mathbf{H}^i(\mathbf{r}) = (H_\theta^i \hat{\theta} + H_\phi^i \hat{\phi}) e^{-j\mathbf{k} \cdot \mathbf{r}} \quad (5.24a)$$

$$\mathbf{E}^i(\mathbf{r}) = (E_\theta^i \hat{\theta} + E_\phi^i \hat{\phi}) e^{-j\mathbf{k} \cdot \mathbf{r}} \quad (5.24b)$$

$$\mathbf{k} = k\hat{k} = -k(\sin \theta_0 \cos \phi_0 \hat{x} + \sin \theta_0 \sin \phi_0 \hat{y} + \cos \theta_0 \hat{z}) \quad (5.24c)$$

As coordenadas θ_0 e ϕ_0 especificam os ângulos a partir dos quais a onda plana incidente chega na esfera e $\hat{\theta}$, $\hat{\phi}$ e $\hat{k}$ são os vetores unitários correspondentes do sistema esférico. O vetor $-\hat{k}$ corresponde ao vetor unitário $\hat{r}$.

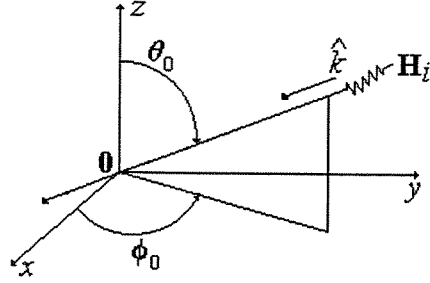


Figura 5.6 – Onda plana incidente e seus componentes.

Se escolhermos corretamente os vetores de base $\hat{u}_n^i$, os elementos da matriz Z_{mn}^{ij} serão independentes da excitação. Assim, precisaremos calcular seus valores apenas uma vez, e para cada nova excitação só é necessário calcularmos a matriz V_m^i e aproveitar os resultados já obtidos, que se manterão inalterados.

Com estes dados é possível calcularmos as seções retas de espalhamento bi-estático de ondas elétricas ou magnéticas, dadas pelas equações 5.24, que são definidas pelas seguintes equações:

$$\begin{aligned}\sigma_{pq}(\theta_i, \phi_i; \theta_e, \phi_e) &= \lim_{R \rightarrow \infty} 4\pi R^2 \frac{|E_q^e(\theta_e, \phi_e)|^2}{|E_p^i(\theta_i, \phi_i)|^2} \\ &= \lim_{R \rightarrow \infty} 4\pi R^2 \frac{|H_q^e(\theta_e, \phi_e)|^2}{|E_p^i(\theta_i, \phi_i)|^2}\end{aligned}\tag{5.25}$$

Aonde R é a distância entre a fonte e o alvo; (θ_i, ϕ_i) e (θ_e, ϕ_e) especificam as direções de propagação da onda incidente e da onda espalhada respectivamente. Os valores de p e q variam conforme a polarização da onda incidente. Por exemplo, para o caso de polarização linear, se tivermos $\sigma_{\theta\theta}$, significa que utilizaremos o componente θ do campo incidente e o

componente ϕ da onda espalhada no cálculo da equação 5.25. Geralmente os parâmetros de maior interesse são $\sigma_{\theta\theta}$ e $\sigma_{\phi\phi}$.

Capítulo 6

Resultados

Neste Capítulo mostraremos os resultados obtidos utilizando-se as técnicas propostas para os três tipos de refinamento: bidimensional, tridimensional superficial e tridimensional volumétrico.

Para o caso bidimensional, primeiro faremos comparações entre o nosso método e alguns dos principais métodos estudados no Capítulo 2 através do refinamento de uma mesma malha inicial para cada caso e analisando os resultados com respeito à qualidade da malha final, tempo de refinamento, etc. Veremos, ainda, alguns exemplos de malhas que podem ser obtidas com o auxílio do nosso programa e, por último, analisaremos dois casos numéricos, para podermos verificar se a malha construída com nosso algoritmo conduz a uma resposta numericamente válida.

Nos refinamentos tridimensionais também mostraremos as malhas obtidas como exemplos do funcionamento de nossos algoritmos e analisaremos suas características. Também apresentaremos um caso numérico para cada tipo de refinamento 3D, superficial e volumétrico.

Começaremos com uma descrição dos recursos utilizados, e programas desenvolvidos durante a execução deste trabalho.

6.1 – Recursos Utilizados

Todos os programas utilizados neste trabalho foram escritos em linguagem Fortran77 [Hehl86] ou Fortran90 [Redwine95]. Optamos pela utilização desta linguagem pois a maioria dos códigos existentes no Departamento já a utilizava. Caso escolhêssemos outra linguagem mais moderna (C++, por exemplo) seria necessário um extenso trabalho de translação de uma para outra, caso quiséssemos adaptar certas rotinas às nossas necessidades. Inicialmente, utilizou-se o compilador da Universidade de Salford, FTN77/386, versão 1.67 para a compilação dos programas e geração dos arquivos executáveis, passando-se depois a usar o compilador Digital Visual Fortran 5.0 [Digital97].

Para a obtenção da maioria dos resultados, foram utilizados computadores tipo PC Pentium Celeron 333 MHz e 32 Mb de memória RAM, operando com sistema operacional Windows 95. Os resultados para o teste numérico da malha tridimensional superficial foram obtidos em estações de trabalho Sun/Sparcsystem do Departamento.

Os programas utilizados para a obtenção dos resultados, além dos algoritmos de refinamento, foram:

- Um programa que associa as condições de contorno e índices de refração à malha bidimensional de elementos finitos. Este programa ainda gera uma figura tipo *postscript* da malha.

Autor: Prof. Hugo E. H. Figueroa

- Um programa de cálculo através do Método dos Elementos Finitos Nodais bidimensional, ou ‘Solver’, que utiliza os métodos descritos e equações descritos nas

Seções 5.1 e 5.2, para a resolução do nosso problema. As matrizes resultantes são calculadas através do método de ‘Sub-Space Iteration’.

Autor: Prof. Hugo E. H. Figueroa

- Um programa de cálculo através do Método dos Momentos, para a resolução de problemas de espalhamento em estruturas tridimensionais superficiais.

Autor: Marcelo I. Davanço.

- Um programa de cálculo através do Método dos Elementos Finitos de Aresta, para a resolução do problema numérico tridimensional volumétrico.

Autor: Antônio M. F. Frasson.

6.2 – Resultados do Refinamento Bidimensional

Nesta Seção apresentaremos alguns resultados para o refinamento bidimensional e comentaremos os mesmos. Primeiramente faremos um estudo comparativo entre algumas técnicas apresentadas no Capítulo 2 para o refinamento bidimensional, incluindo o método proposto por nós na Seção 2.3.3.

Como segunda demonstração da funcionalidade de nosso algoritmo, mostraremos algumas malhas obtidas com o auxílio de nosso algoritmo. Cada malha destas possui características específicas e são analisadas para estudar o comportamento do algoritmo de refinamento em relação a estas características.

Finalmente, mostramos na prática o uso de malhas adaptativas no cálculo através do método dos elementos finitos nodais, com a análise modal de dois casos distintos.

6.2.1 – Estudo Comparativo

Como dito anteriormente, nesta seção faremos uma comparação entre os resultados obtidos com nosso algoritmo e aqueles obtidos com implementações baseadas nas descrições dos algoritmos mostrados no Capítulo 2. Recapitulando, os métodos são: Baricêntrico, Bissecção pela maior aresta de Nambiar e Bissecção com troca de diagonais de Rivara. O outro método apresentado por ela, que utiliza o método LEPP sem a troca de diagonais, produz resultados extremamente semelhantes aos obtidos com a técnica de Bissecção com lista ordenada, tendo como diferença apenas o tempo de processamento, que foi melhorado com a implantação da técnica LEPP no lugar da lista ordenada. Após isto destacaremos as diferenças entre cada resultado e aquele obtido por nós.

Para manter o estudo confiável, foram utilizadas as mesmas condições em todos os casos. A malha inicial usada em todos os refinamentos possui 6 elementos e 7 nós, conforme Figura 6.1. Propositadamente, escolhemos esta malha com poucos nós e com alguns elementos com aspecto ruim com a finalidade de analisar a capacidade do algoritmo em gerar rapidamente uma malha de boa qualidade a partir de uma malha rudimentar, corrigindo eventuais elementos ruins da malha original. A qualidade dos piores elementos, as cunhas, é de 0,9, enquanto que a qualidade máxima é de 0,90. Estes valores foram obtidos utilizando a Equação 2.1. Os ângulos máximo e mínimo são de 90,0 e 6,34 graus, respectivamente.

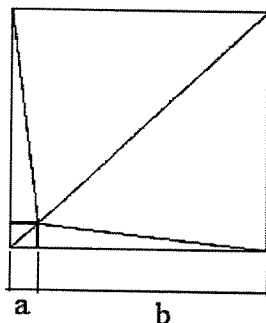


Figura 6.1 – Malha utilizada no estudo comparado de refinamento.

A malha representa um guia de onda tipo canal. Devido à simetria do mesmo, o refinamento pode ser feito em um quarto da estrutura. Como neste caso estamos estudando a qualidade da malha e rapidez do programa, as dimensões da estrutura não são relevantes, bastando indicar que a largura total da estrutura, $a + b$, é dez vezes maior que a largura do canal interno, a .

Para monitorar o refinamento da malha, utilizamos funções de densidade pré-definidas uma vez que não seria utilizada a resolução através de elementos finitos para utilizarmos o controle de erro para guiar o refinamento. Nos exemplos mostrados no estudo comparado, utilizamos uma função gaussiana centrada no canal interno. O número final de elementos, estabelecido como limite para o término do refinamento, foi ao redor de 3000 elementos. Isto foi estabelecido para podermos analisar o comportamento dos algoritmos quando necessitamos uma grande quantidade de elementos.

Em cada caso, mostraremos a malha obtida e um detalhe da região central da mesma para a análise visual da qualidade da malha. Também mostraremos gráficos da distribuição da qualidade dos elementos e dos ângulos internos dos elementos em função das quantidades normalizadas de elementos e ângulos internos, respectivamente. O primeiro permite analisar a qualidade geral da malha final obtida enquanto que o segundo gráfico nos permite analisar a distribuição dos ângulos internos da malha e, conseqüentemente, a qualidade da mesma.

6.2.1.1 – Refinamento Baricêntrico

Usando a malha inicial e os dados descritos acima, geramos a malha mostrada na Figura 6.2(a), com uma implementação do algoritmo de refinamento baricêntrico apresentado na Seção 2.3.1. Nesta malha, após terminado o processo de refinamento, foi aplicada uma suavização recursiva, até que nenhuma mudança significativa na posição dos nós ocorresse (em torno de 10%). A malha obtida possui 3012 elementos e 1551 nós. Na Figura 6.2(b), mostramos um detalhe da região do canal, que está densamente refinada. Sua

qualidade varia de 0,57 até 0,999, e seus ângulos mínimo e máximo são 23,62 e 122,55 graus, respectivamente.

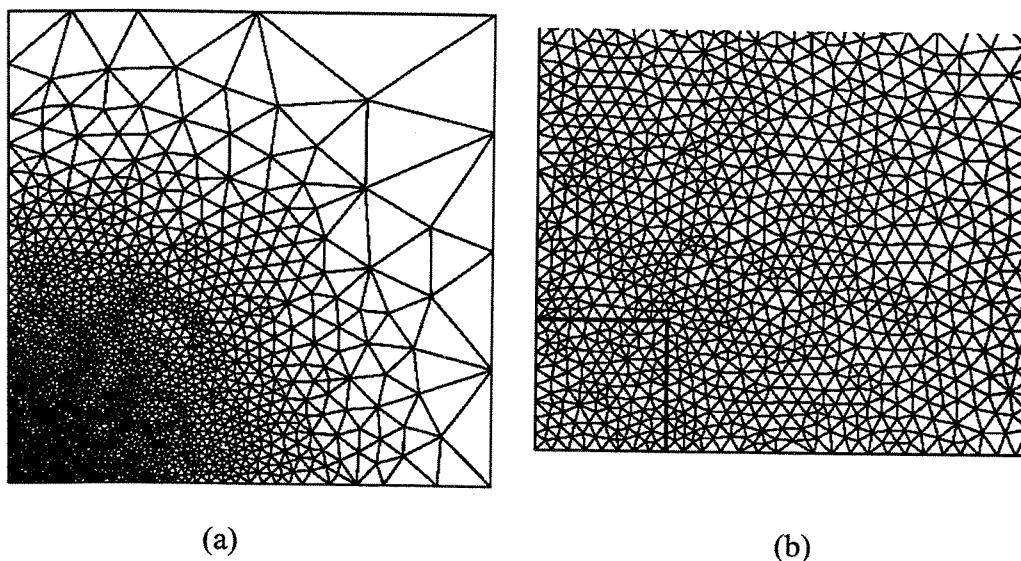


Figura 6.2 – Refinamento utilizando a técnica baricêntrica.(a) Malha resultante e (b) detalhe da região do canal.

No gráfico da Figura 6.3(a), mostramos a qualidade dos elementos versus o número normalizado de elementos. A partir dele podemos concluir que aproximadamente 90% dos elementos possui uma qualidade acima de 0,9, e ao redor de 98% dos elementos acima de 0,8, indicando, portanto, uma malha de excelente qualidade.

Na Figura 6.3(b), vemos a distribuição dos ângulos internos da malha apresentada na Figura 6.2. Como já havíamos visto anteriormente no gráfico da Figura 6.3(a), esta malha possui uma boa qualidade geométrica, pois como podemos concluir aproximadamente 65% dos ângulos internos estão próximos a 60 graus, o que denota a existência de vários triângulos equiláteros.

Como vimos anteriormente, o ângulo mínimo da malha inicial era de 6,34 graus, enquanto que o ângulo mínimo da malha final foi de 23,62. Isto comprova a afirmação feita

na Seção 2.3.1 de que o uso da troca de diagonais aumenta o ângulo mínimo de um conjunto de elementos.

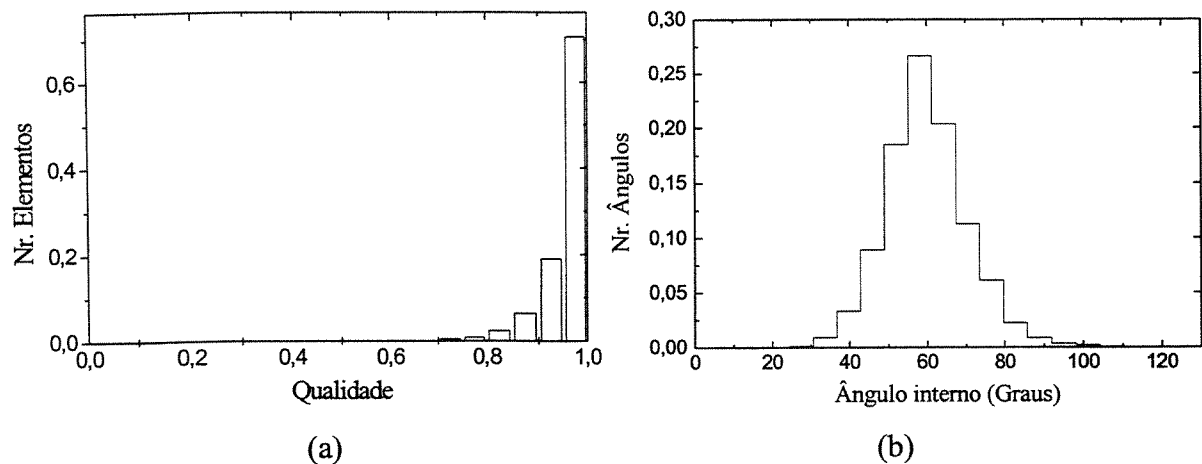


Figura 6.3 – Gráficos das distribuições de (a) qualidade e (b) ângulos internos em função das quantidades normalizadas para a malha apresentada na Figura 6.2.

As desvantagens desta malha são, obviamente, a necessidade do uso da técnica de relaxação, que aumenta o tempo de processamento em até 50%, e o surgimento de frentes de elementos finitos, que foram atenuadas com o uso da técnica de relaxação. Também podemos concluir que, devido à alta variação no tamanho dos elementos, na forma 1:3, o refinamento pode ocorrer mais fortemente em regiões indesejáveis, além de produzirem uma certa irregularidade na malha final.

6.2.1.2 – Refinamento pela Bissecção da Maior Aresta - Nambiar

Utilizando-se uma implementação da técnica de refinamento pela bissecção da maior aresta, na aproximação descrita por [Nambiar93], chegamos à malha final mostrada na Figura 6.4(a). Nesta malha obtivemos 3039 elementos e 1545 nós. Na Figura 6.4(b), vemos o detalhe da região de maior refinamento. A qualidade da malha obtida varia de 0,127 até 0,959, enquanto que os ângulos internos variam de 6,34 a 167,32 graus.

É possível notar visualmente, neste caso, que devido ao uso de uma malha inicial relativamente ruim, a malha final possui vários elementos de aspecto pobre, o que denota a forte vinculação entre a qualidade da malha resultante e a qualidade dos elementos da malha inicial, como havíamos dito na Seção 2.3.2. Como podemos ver na malha final, e no detalhe, aqueles elementos quasi-equiláteros da malha inicial geraram elementos de bom formato geométrico, enquanto que a incidência dos elementos mal formados ocorreu aonde os elementos tipo ‘cunha’ estavam.

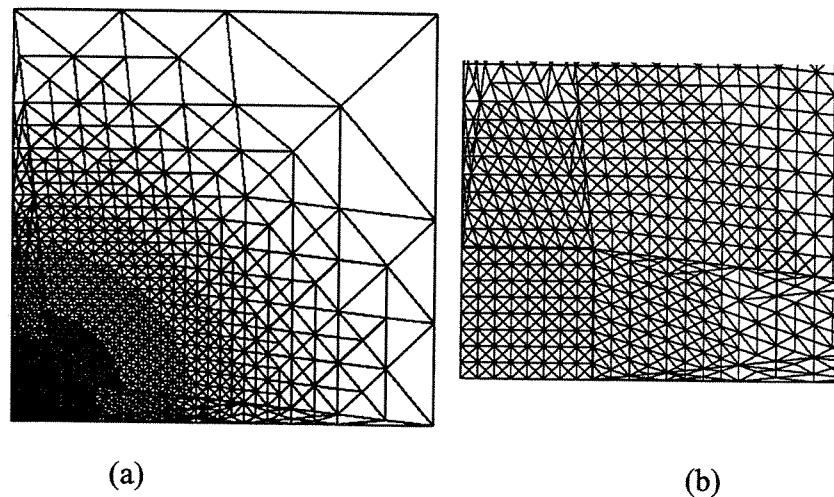


Figura 6.4 – Refinamento utilizando a técnica de bissecção com lista ordenada. (a) Malha final e (b) detalhe do canal interno.

Na Figura 6.5(a) mostramos a distribuição de qualidade da malha versus o número normalizado de elementos. Neste caso, o refinamento produziu aproximadamente 75% de elementos acima da qualidade 0,8. Mesmo com a ocorrência de elementos com qualidade inferior a 0,3, aproximadamente 6%, podemos considerar que esta malha irá produzir bons resultados, devido à alta incidência de elementos com bom formato geométrico.

Embora visualmente a malha aparente certas características semelhantes às chamadas ‘frentes de elementos’, uma análise cuidadosa nos mostra que a variação não é tão abrupta, sendo da forma 1:2.

Na Figura 6.5(b) vemos a distribuição dos ângulos internos em relação ao número normalizado de ângulos internos. Como podemos ver, a grande maioria dos ângulos internos está concentrada entre 30 e 100 graus, o que mostra a existência de poucos elementos eqüiláteros (com todos ângulos internos iguais a 60 graus), mas com uma grande quantidade de triângulos retângulos e/ou semi-eqüiláteros com ângulos internos variando ao redor dos picos em 45 e 100 graus .

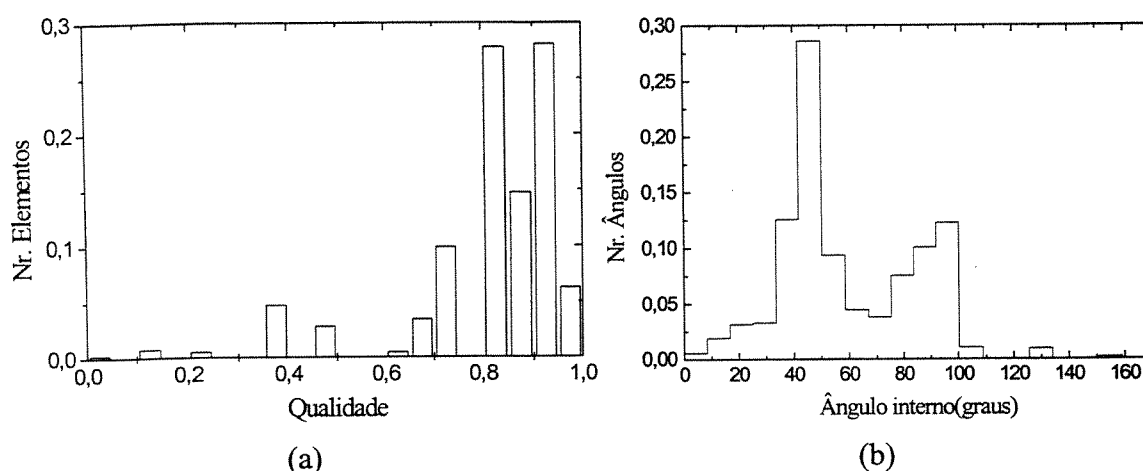


Figura 6.5 – Distribuição da (a) qualidade e (b) dos ângulos internos da malha final, mostrada na Figura 6.4.

O ângulo mínimo da malha final foi igual ao ângulo mínimo da malha inicial, o que vai de encontro à previsão feita na Seção 2.3.3.1, aonde dissemos que o ângulo mínimo final seria igual ou superior à metade do ângulo mínimo inicial.

Como dissemos, esta técnica produz elementos conformes e de bom aspecto geométrico, mas a qualidade final está fortemente ligada à qualidade inicial da malha. Além disso, a necessidade de uma lista ordenada de elementos torna este algoritmo extremamente demorado, como veremos no final do estudo comparado, o que é extremamente indesejável.

6.2.1.3 – Refinamento pela Bissecção da Maior Aresta – Rivara

Nesta seção, mostraremos as malhas obtidas a partir da malha inicial mostrada na Figura 6.1 com o auxílio da implementação do algoritmo apresentado por Rivara, que utiliza o método LEPP em conjunto com a retriangulação Delaunay.

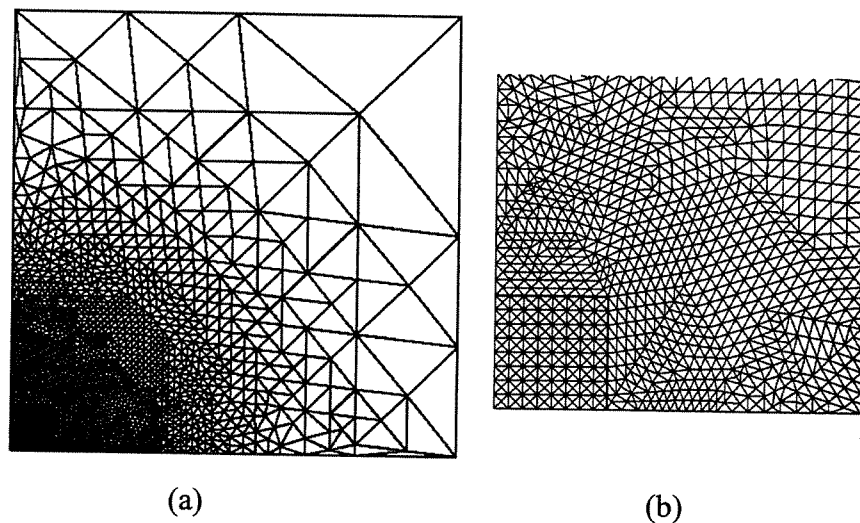


Figura 6.6 – Refinamento utilizando a técnica de Bissecção+LEPP com troca de diagonais. (a) Malha final e (b) detalhe da região central.

Após o refinamento, obtivemos a malha mostrada na Figura 6.6(a), com 2968 elementos e 1531 pontos. No detalhe da Figura 6.6(b) vemos a região central. A malha resultante tem um ângulo mínimo de 6,34 graus e um ângulo máximo de 167,31 graus. A qualidade da malha final varia de 0,127 a 0,999. Obviamente os elementos com os piores valores de qualidade são os triângulos que possuem ângulos extremamente abertos e/ou fechados, situados no canto inferior direito e no canto superior esquerdo. Estes elementos não foram afetados pela troca de diagonais por estarem muito distantes da região de refinamento e situados em uma fronteira externa.

Na Figura 6.7(a) vemos o gráfico da qualidade da malha resultante apresentada na Figura 6.6. Como podemos concluir, 94% dos elementos da malha possuem qualidade superior a 0,8 e somente 0,3% dos elementos estão abaixo do fator de qualidade 0,4. Como

estes elementos ruins estão longe da região central, onde os campos eletromagnéticos se concentram, estes elementos pobres não afetarão o resultado de uma simulação numérica.

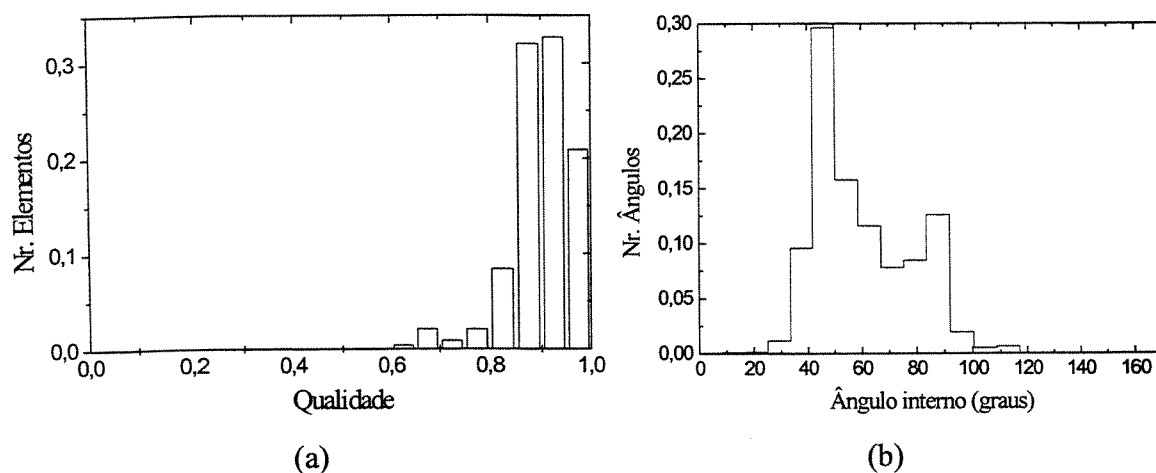


Figura 6.7 – Distribuição da (a) qualidade e (b) dos ângulos internos da malha final, mostrada na Figura 6.6.

Analisando o gráfico da distribuição de ângulos internos, Figura 6.7(b), vemos que, com o uso da troca de diagonais, a quantidade de ângulos iguais a 60 graus aumentou em relação aos resultados obtidos com a técnica de bissecção com lista ordenada, embora a distribuição geral não seja tão boa quanto aquela obtida com a técnica baricêntrica, apresentada na Figura 6.3(b). Esta melhora é atribuída à retriangulação Delaunay, que acaba elevando o ângulo mínimo dos elementos envolvidos, embora, neste caso específico, isto acabou não ocorrendo. Isto ocorreu devido ao fato de que os elementos com ângulos iguais ao ângulo mínimo inicial estão situados em uma posição distante da região de refinamento e em uma fronteira externa. Da mesma forma que ocorreu no caso da bissecção mais lista ordenada (Seção anterior), os ângulos internos se concentram entre 35 e 90 graus, o que denota a presença de muitos triângulos semi-equiláteros e/ou retângulos.

Como dissemos na Seção 2.3.2, este algoritmo é rápido e produz elementos conformes e de bom formato geométrico, gerando malhas conformes de boa qualidade. O uso da técnica LEPP contorna parcialmente a vinculação da qualidade final da malha em

relação à qualidade inicial, restando apenas alguns poucos elementos em situações especiais, conforme foi mostrado.

6.2.1.4 – Refinamento pela Técnica Proposta.

Nesta seção apresentamos os resultados para o refinamento da malha inicial padrão já apresentada, utilizando o algoritmo por nós proposto na Seção 2.3.3. A malha final obtida é mostrada na Figura 6.8(a), enquanto que o detalhe da região do canal está na Figura 6.8(b). A malha final possui 3001 elementos e 1545 nós. O ângulo interno mínimo é de 19,1 graus e o máximo é de 128,9 graus, enquanto que a qualidade varia de 0,471 até 0,999.

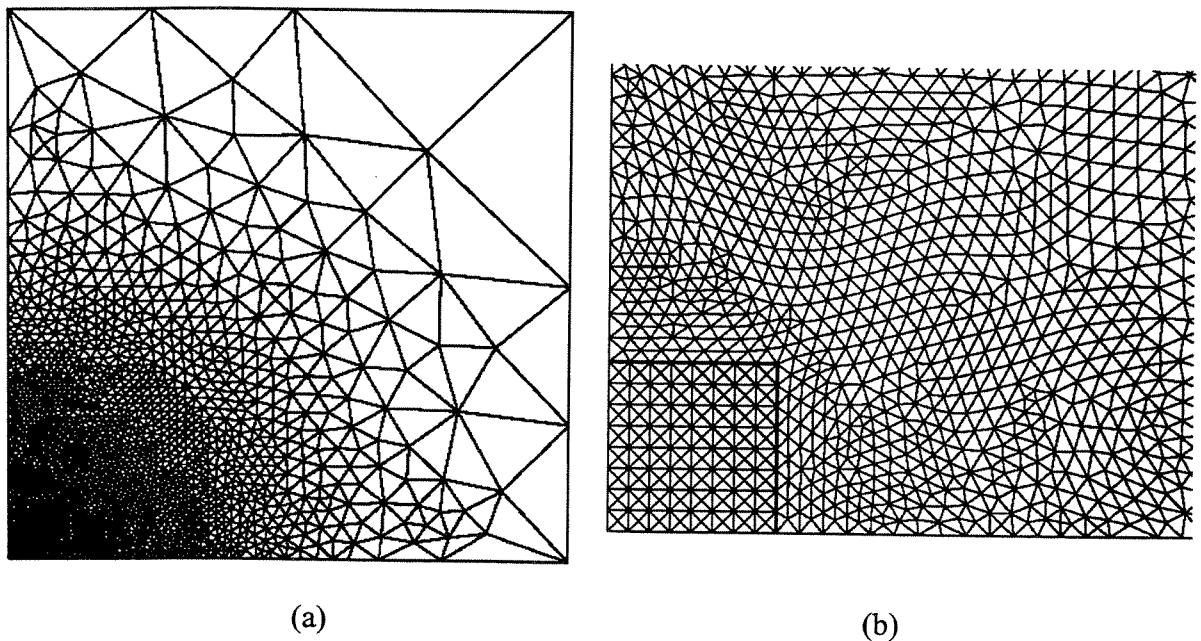


Figura 6.8 – Refinamento utilizando a técnica proposta. (a) Malha final e (b) detalhe.

Na Figura 6.9(a) vemos o gráfico da qualidade da malha final. Analisando-o mais detidamente, notaremos que aproximadamente 97% dos elementos possuem qualidade acima de 0,8, sendo que o triângulo de pior qualidade tem um valor acima de 0,4, mostrando que esta malha possui uma qualidade excelente. Isto também pode ser visto se

analisarmos o gráfico da Figura 6.9(b), onde traçamos a distribuição dos ângulos internos na malha. O gráfico nos mostra que aproximadamente 53% dos ângulos da malha estão entre 50 e 70 graus, denotando boa quantidade de triângulo equiláteros.

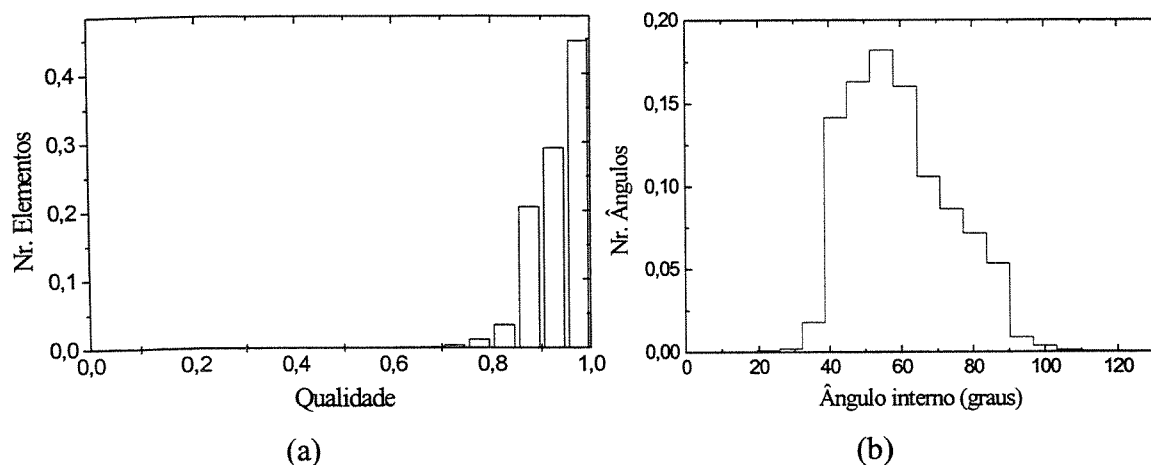


Figura 6.9 – Distribuição (a) da Qualidade e (b) dos ângulos internos da malha final.

Como foi observado, o refinamento através do método proposto conjuga várias técnicas (bissecção dos elementos, LEPP, troca de diagonais e relaxação), e o resultado é uma malha de boa qualidade, conforme podemos notar. O uso da técnica da bissecção dos elementos proporciona uma transição suave e conforme entre os elementos, enquanto que o uso da busca localizada eliminou a necessidade da lista ordenada e diminui o tempo de processamento. O uso da troca de diagonais e da relaxação dos nós aumenta a qualidade total da malha e eleva o ângulo mínimo da malha, melhorando a qualidade dos elementos e praticamente eliminando os triângulos de aspecto geométrico pobre.

6.2.1.5 – Comparações entre os resultados

Nesta Seção faremos comparações entre os resultados dos refinamentos com outros algoritmos e com o algoritmo proposto por nós. Compararemos a qualidade das malhas, surgimento de frentes de elementos finitos, quantidade de ângulos muito abaixo ou muito acima de 60 graus (considerado o caso ideal) e também o tempo de computação.

Compararemos, primeiramente, os tempos de computação, mostrados no gráfico da Figura 6.10(a). Os resultados das medições de tempo estão numerados de acordo com sua apresentação neste estudo, assim o tempo número 1 corresponde ao tempo de refinamento do algoritmo baricêntrico, 2 à técnica de refinamento através da bissecção de Nambiar, 3 ao segundo método apresentado por Rivara (Bissecção com troca de diagonais) e 4 refere-se à técnica de refinamento por nós proposta. Pelo primeiro gráfico podemos notar que o tempo de execução do algoritmo de bissecção de Nambiar possui um tempo de computação muito acima dos outros, devido ao uso da lista ordenada, e possui um comportamento semelhante a uma função de segundo grau com relação ao número de elementos.

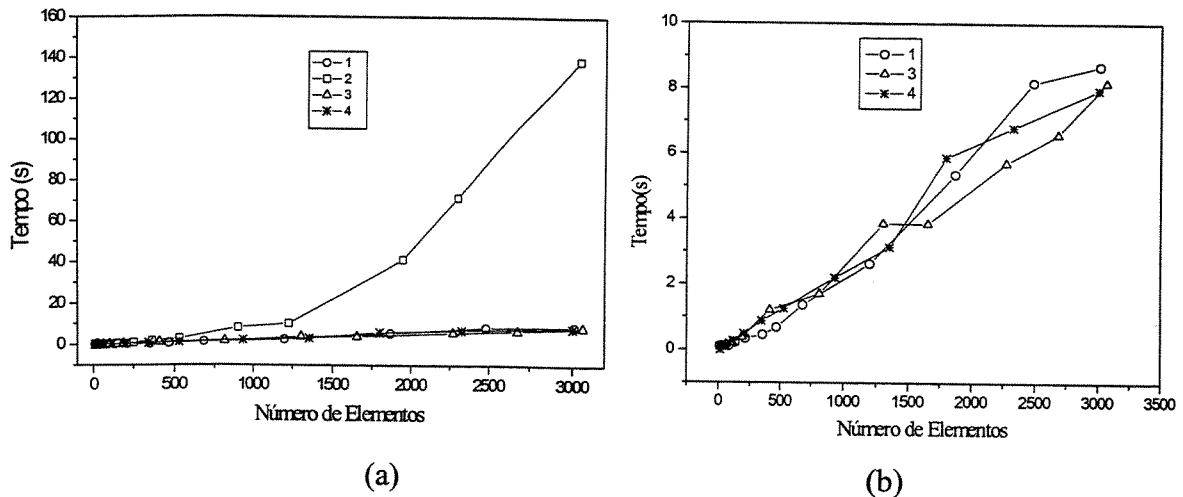


Figura 6.10 – Comparação entre os tempos de execução das técnicas apresentadas: 1 – Técnica baricêntrica, 2 – Algoritmo de bissecção + lista ordenada, 3 – Algoritmo de bissecção + LEPP +troca de diagonais e 4 – Técnica proposta.

No gráfico 6.10(b), mostramos um detalhe da comparação entre os tempos, mostrando apenas aqueles de menor tempo de execução (isto é, as técnicas 1, 3 e 4). Podemos ver que o comportamento deles é praticamente linear, o que já era esperado pois a inserção de novos nós acontece de forma constante [Rivara96]. É possível notar que todas as técnicas possuem tempos de execução bastante similares entre si, com pequenas flutuações que ocorreram provavelmente devido a pequenos problemas de 'hardware'.

Comparando nosso algoritmo com o algoritmo baricêntrico, vemos que os tempos de processamento da técnica baricêntrica e da técnica proposta são similares. A técnica baricêntrica gera triângulos com qualidade e ângulos internos um pouco melhores que aqueles obtidos pelo nosso algoritmo. Entretanto nosso algoritmo se sobressai no aspecto final da malha, sem a ocorrência de frentes de elementos e com o refinamento ocorrendo apenas nas regiões de interesse, sem o surgimento de irregularidades. Convém ressaltar aqui que a relaxação dos nós na técnica baricêntrica foi feita até que a mudança na posição dos nós seja 100 vezes menor que a mudança inicial. Se quisermos atingir valores melhores, conseqüentemente o tempo de execução também aumentará. Nosso algoritmo produz portanto resultados, no pior dos casos, semelhantes àqueles obtidos com uso da técnica baricêntrica.

A comparação de nosso algoritmo com o algoritmo de refinamento através da bissecção da maior aresta de Nambiar nos mostra que nossa aproximação possui diversas vantagens em relação a ele, tanto em relação à qualidade de elementos e ângulos internos, quanto ao tempo de execução. Como podemos verificar nas malhas para cada caso, Figuras 6.4 e 6.8, o uso de nosso algoritmo evita a vinculação da malha final em relação à malha inicial, melhorando inclusive o valor do ângulo interno mínimo.

Finalmente, confrontando a nossa proposta com a técnica de bissecção mais troca de diagonais, mostrada por Rivara, vemos que ambas produzem bons resultados em tempos equivalentes, mas nosso algoritmo se sobressai na qualidade dos elementos e ângulos internos, pois devido ao uso da relaxação final, os casos aonde a utilização da troca de diagonais não solucionava (elementos-cunha mostrados na Figura 6.6(a)), são suavizados com uso de nossa técnica, aumentando a qualidade dos elementos e elevando o ângulo mínimo da malha.

6.2.2 – Exemplos de Refinamento

Nesta Seção, apresentamos dois exemplos de malhas obtidas com o auxílio de nosso algoritmo. Em todos os casos, partimos de uma malha com poucos elementos, até chegarmos a uma malha com uma quantidade razoável de elementos. Em todos os casos, mostraremos a malha inicial, a malha resultante e a qualidade final da malha. Cada caso foi estudado por possuir certas peculiaridades que nos ajudam a avaliar a robustez de nosso algoritmo. Novamente nestes casos não foi conduzida a análise modal de cada exemplo, o que será feito na Seção 6.2.3. As dimensões reais de cada exemplo foram omitidas pois estamos interessados apenas na análise da funcionalidade do algoritmo e das características geométricas das malhas resultantes.

6.2.2.1 – Refinamento de um guia tipo microfita

No primeiro exemplo, mostramos o refinamento de um guia tipo microfita. Este exemplo é interessante, pois mostra a capacidade de nosso algoritmo em tratar de malhas com cavidades internas e grande variação no tamanho das arestas. Este guia é composto de uma fina região condutora central colocada sobre um substrato dielétrico. O material da camada superior é o ar. Devido à fina região condutora central, ocorre uma grande variação no aspecto dos elementos. As dimensões típicas de um guia deste tipo são: largura a , altura $a/2$, altura do dielétrico $a/4$ e largura da região condutora b [Jin93], conforme vemos na Figura 6.11.

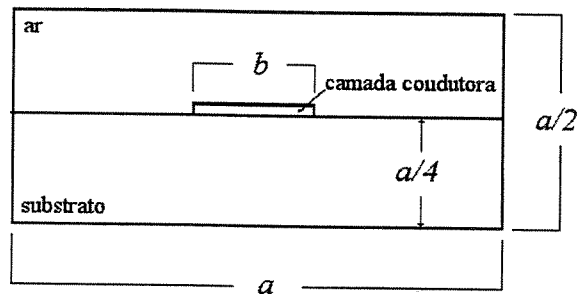


Figura 6.11 – Estrutura de um guia de ondas tipo microfita blindada.

Na Figura 6.12, vemos a malha inicial utilizada para o refinamento. Esta malha possui apenas 16 elementos e 16 nós. Como dissemos, devido à fina região central, esta malha possui alguns elementos que apresentam uma variação muito grande em suas dimensões, implicando em uma qualidade muito ruim. Tais elementos são mostrados nas regiões centro-laterais da Figura 6.12. A malha inicial possui um ângulo mínimo de 2,29 graus e um ângulo máximo de 92,33 graus, enquanto que a sua qualidade varia de 0,069 até 0,866.

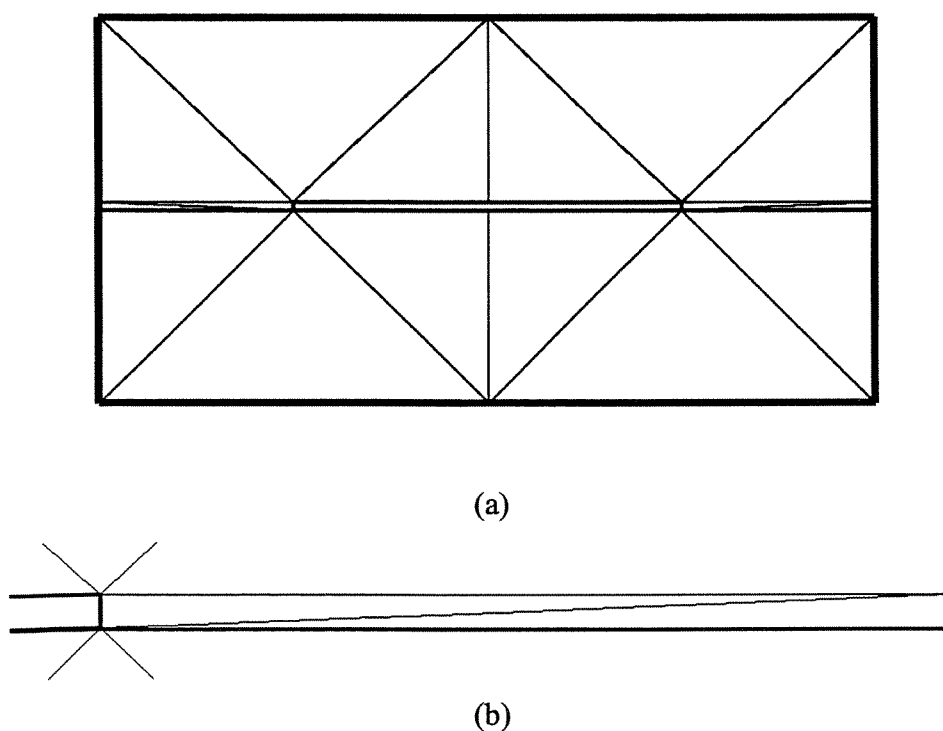
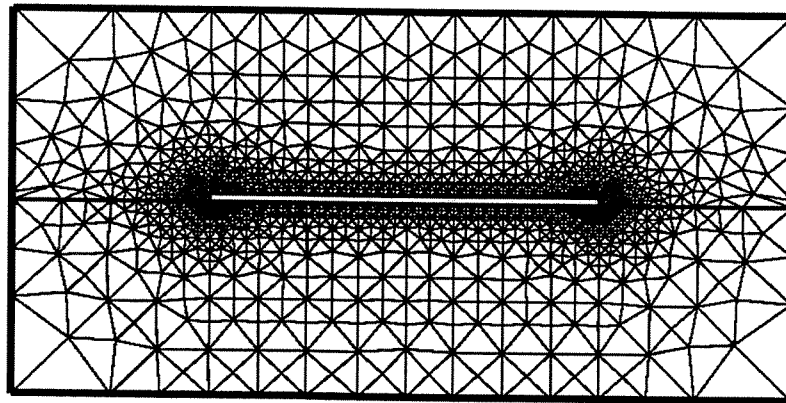


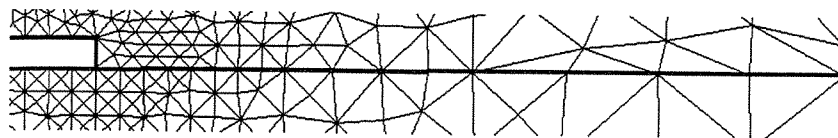
Figura 6.12 – (a) Malha inicial para o guia tipo microfita e (b) um detalhe de uma das regiões centro-laterais que contém os elementos de pior aspecto.

A função de densidade utilizada produziu um refinamento mais acentuado ao redor e nos cantos da tira central. Na Figura 6.13 vemos o resultado final do refinamento. Nesta malha obtemos um total de 2534 elementos e 1353 pontos nodais. O ângulo mínimo e máximo foram de 10,86 e 111,86, respectivamente. A qualidade mínima obtida foi de 0,30 e a máxima foi de 0,99. Isto demonstra que nosso algoritmo foi capaz de contornar a

existência daqueles elementos ruins mostrados na malha inicial, e melhorar a qualidade e o ângulo mínimo da malha final.



(a)



(b)

Figura 6.13 – (a) Malha resultante do refinamento do guia tipo microfita, Figura 6.11 e (b) detalhe da região centro-lateral, mostrada na figura 6.12(b), após o refinamento.

Nas Figuras 6.14(a) e (b) vemos os gráficos da distribuição de qualidade e dos ângulos internos. É possível deduzir que 96% dos elementos possuem qualidade superior a 0,8, e apenas 0,0007% dos elementos estão abaixo de 0,4. Isto demonstra a boa qualidade da malha obtida, apesar da malha possuir uma pequena quantidade de elementos com ângulo próximos a 60 graus. Isto se deve ao fato, já mostrado em exemplos anteriores, da existência de triângulos retângulo e/ou semi-equiláteros na malha. Pela Figura 6.14(b) é possível notar ainda a baixa quantidade de ângulos inferiores a 30 graus e superiores a 100, o que demonstra a quase inexistência de triângulos obtusos.

Este exemplo mostra a capacidade de nosso algoritmo em lidar com elementos ruins em malhas com cavidades internas. Como vimos, o método produz malhas compatíveis e concordantes, eliminando os elementos de aspecto ruim da malha inicial e melhorando a qualidade final da malha.

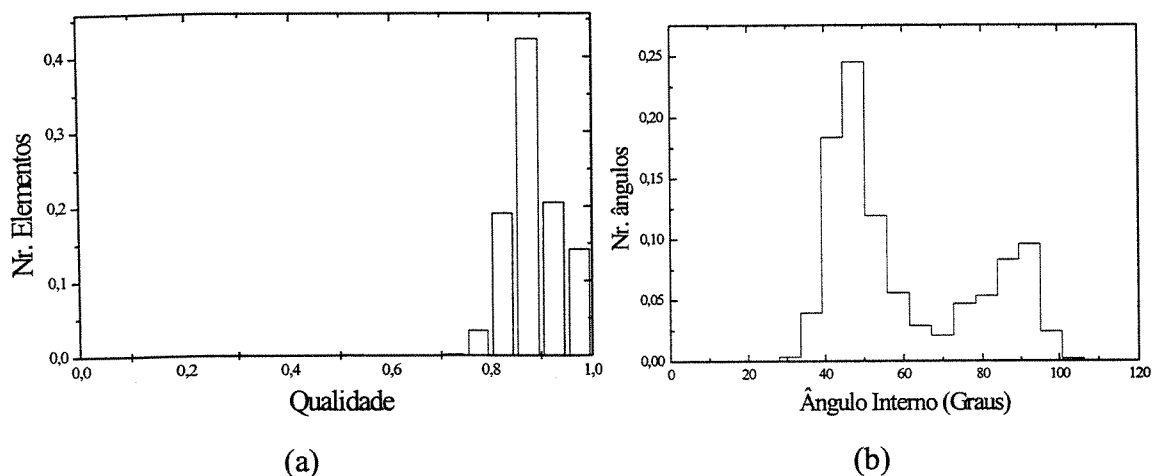


Figura 6.14 – Gráficos das distribuições (a) da qualidade em relação ao número normalizado de elementos e (b) dos ângulos internos em relação à quantidade normalizada de ângulos internos.

6.2.2.2 – Refinamento de um Acoplador óptico

Neste exemplo, mostraremos o refinamento de um acoplador baseado em fibras ópticas fundidas. Este exemplo é interessante pelo fato da malha possuir interfaces internas curvas que devem ser mantidas. Assim, quando tivermos um ponto que deva ser inserido nestas interfaces, este ponto deverá ficar equidistante dos dois pontos da aresta selecionada para subdivisão e ainda verificar a equação da curva, que é definida pelo usuário, conforme Seção 2.3.3.1. O acoplador óptico é composto de dois núcleos de raio r e índice de refração n_1 situados a uma distância d um do outro, envoltos por um substrato circular de índice n_2 (menor que n_1) e raio R . Este substrato está circundado por uma camada de ar, conforme Figura 6.15.

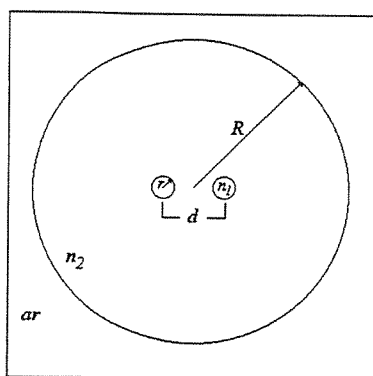


Figura 6.15 – Estrutura de um acoplador a fibra óptica.

Devido às pequenas dimensões dos núcleos em relação ao substrato da fibra, este problema também apresenta grandes variações no tamanho dos elementos. Na Figura 6.16 vemos a malha inicial, que possui 84 elementos e 47 pontos. Seu ângulo mínimo é de 11,72 graus e o máximo é de 128,10. A qualidade da malha inicial varia de 0,337 a 0,963. Os elementos com pior aspecto geométrico estão, claramente, ao redor dos núcleos centrais.

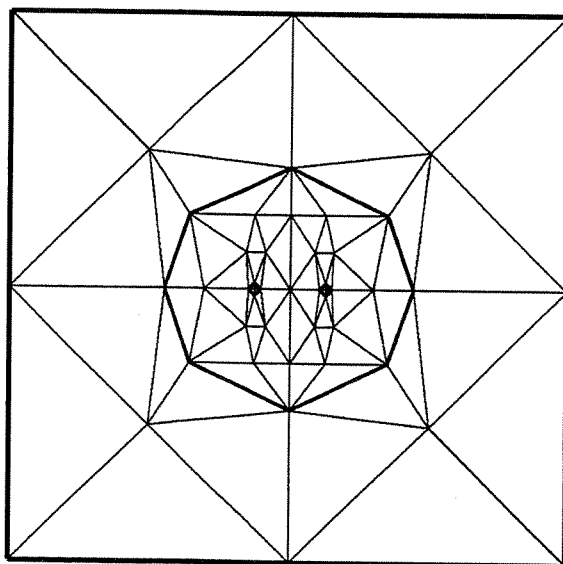


Figura 6.16 – Malha inicial utilizada para o acoplador óptico.

Neste caso, as funções de densidade foram escolhidas de forma a refinar mais ao redor das interfaces e na região central. Estas funções de densidade são formadas por diversas gaussianas, definidas de acordo os dados fornecidos pelo usuário, e são explicados detalhadamente no Apêndice A.2. Na Figura 6.17 mostramos a malha final, contendo 3248

elementos e 1639 pontos. Esta malha possui ângulos internos mínimo e máximo de 25,70 e 111,78 respectivamente. Sua qualidade varia de 0,672 a 0,999.

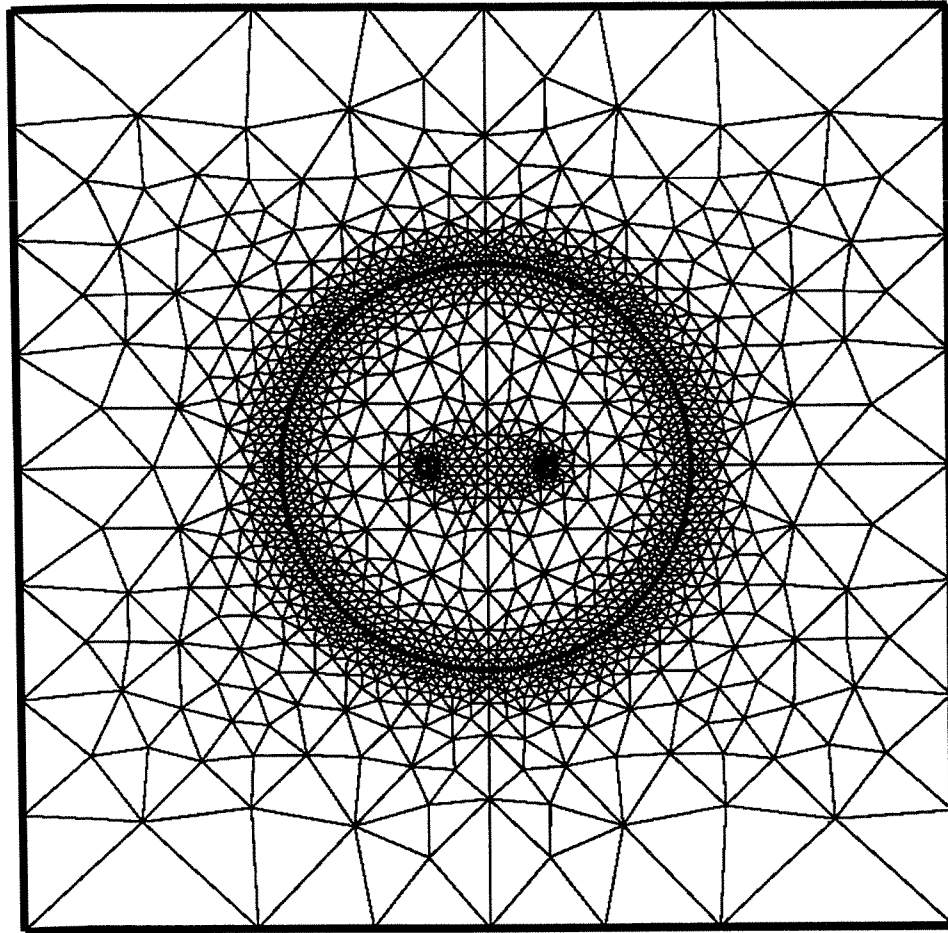


Figura 6.17 – Malha final para o refinamento adaptativo de um acoplador óptico.

Na Figura 6.18 vemos um detalhe da região central aonde é possível ver claramente que o método adapta os pontos para que estes verifiquem a equação da curva que é definida pelo usuário. Também é possível notar a eliminação dos elementos ruins que antes existiam nesta região.

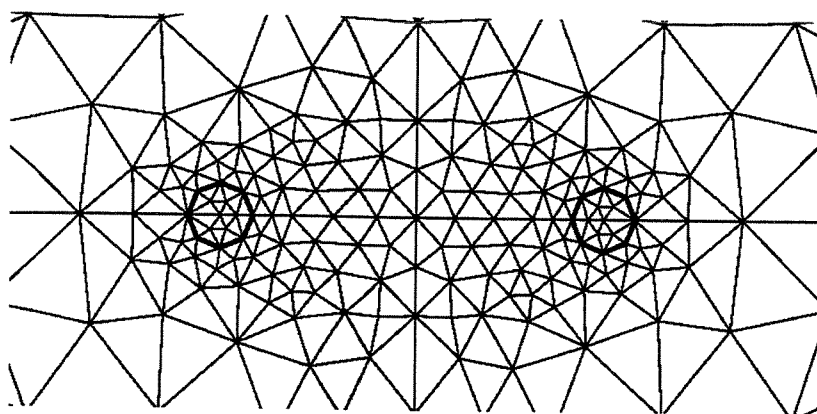


Figura 6.18 – Detalhe da região central da Figura 6.17

Na Figura 6.19, vemos também as distribuições de qualidade e dos ângulos internos em relação às quantidades normalizadas de elementos e ângulos internos, respectivamente. Por eles é possível deduzir que a malha obtida possui grande quantidade de elementos de bom aspecto geométrico, pois 91% dos elementos possuem qualidade igual ou superior a 0,8, sem a ocorrência de elementos com qualidade inferior a 0,4.

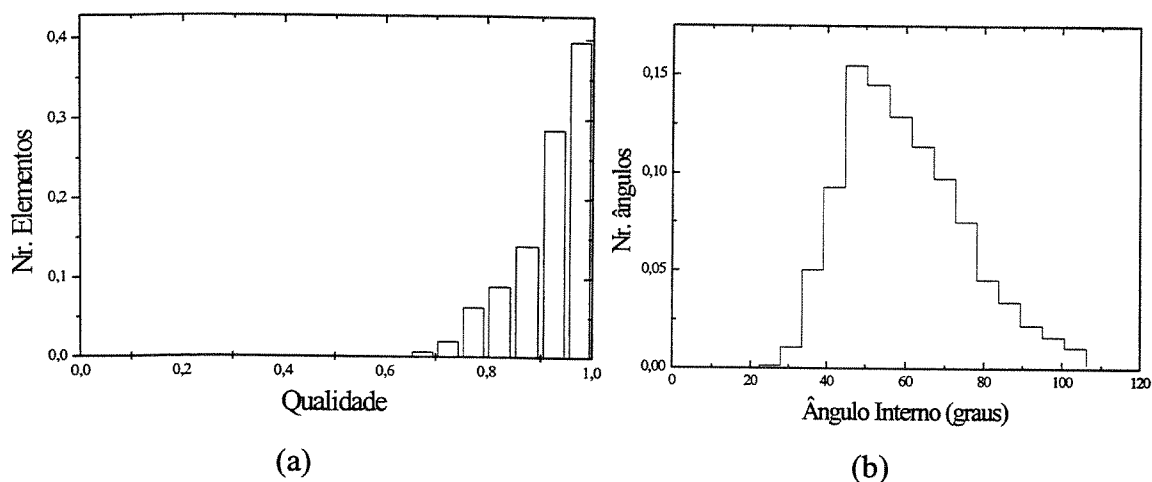


Figura 6.19 – Gráficos das distribuições (a) da qualidade em relação ao número normalizado de elementos e (b) dos ângulos internos em relação à quantidade normalizada de ângulos internos.

A distribuição de ângulos internos também nos mostra a grande incidência de triângulos eqüiláteros, semi-eqüiláteros e/ou retângulos.

Neste exemplo estudamos o comportamento do algoritmo na presença de interfaces curvas. Os resultados obtidos mostram a elevação da qualidade final da malha, além da maximização do ângulo mínimo inicial.

6.2.3 – Resultados Numéricos

Após vários resultados, em que apresentamos apenas a malha e suas características geométricas de qualidade, distribuição de ângulos internos e adaptatividade, veremos agora o estudo de dois casos nos quais faremos o refinamento adaptativo através de estimativas de erro, conforme foi apresentado na Seção 2.2. Veremos como se comporta a malha em situações de análise modal através de elementos finitos, e compararemos os resultados obtidos com aqueles apresentados na literatura, para uma correta avaliação do nosso programa.

6.2.3.1 – Análise de um Guia de Onda tipo Costela ('Rib')

Para confirmar a eficácia das malhas triangulares produzidas com o auxílio do nosso método, na análise modal de guias de onda, primeiramente consideraremos o caso de um guia de onda óptico tipo costela ('rib'). O formato da estrutura e as dimensões (em micrômetros) utilizadas nos cálculos são mostradas na Figura 6.20. Os índices de refração do núcleo n_1 e do substrato n_2 são de 3.44 e 3.34, respectivamente. O comprimento de onda utilizado foi de 1,55 μm e o modo calculado corresponde ao modo TE fundamental.

A Equação 5.4 foi usada para o cálculo através de elementos finitos, conforme mostramos na Seção 5.2.1, na qual nosso autovalor é β^2 , e para isso utilizamos elementos lineares na sua resolução, sendo portanto esperado um erro final na ordem da quarta casa decimal.

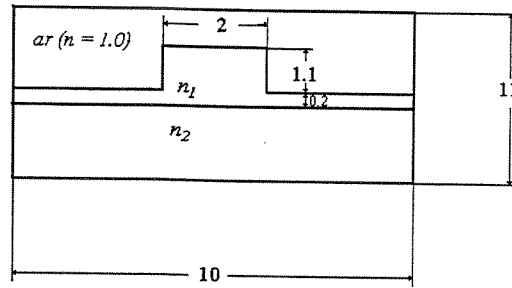


Figura 6.20 – Guia de onda tipo costela e suas dimensões, em micrômetros.

Utilizamos o método das estimativas de erro, Seção 2.2, para monitorar o refinamento da estrutura. O critério de parada estabelecido para terminar o refinamento foi de que o erro global relativo, Equação 2.12, deveria ficar abaixo de 0,1% e o erro admissível final, Equação 2.10, deveria ser menor que 1% do valor do erro admissível inicial. Como critério de comparação utilizamos uma solução obtida através da malha mostrada na Figura 6.21(b), gerada a partir da malha inicial da Figura 6.21(a). Como podemos ver, esta malha que irá determinar os locais onde o refinamento é mais importante foi obtida através do refinamento monitorado por uma função gaussiana com origem no centro da região de guiamento.

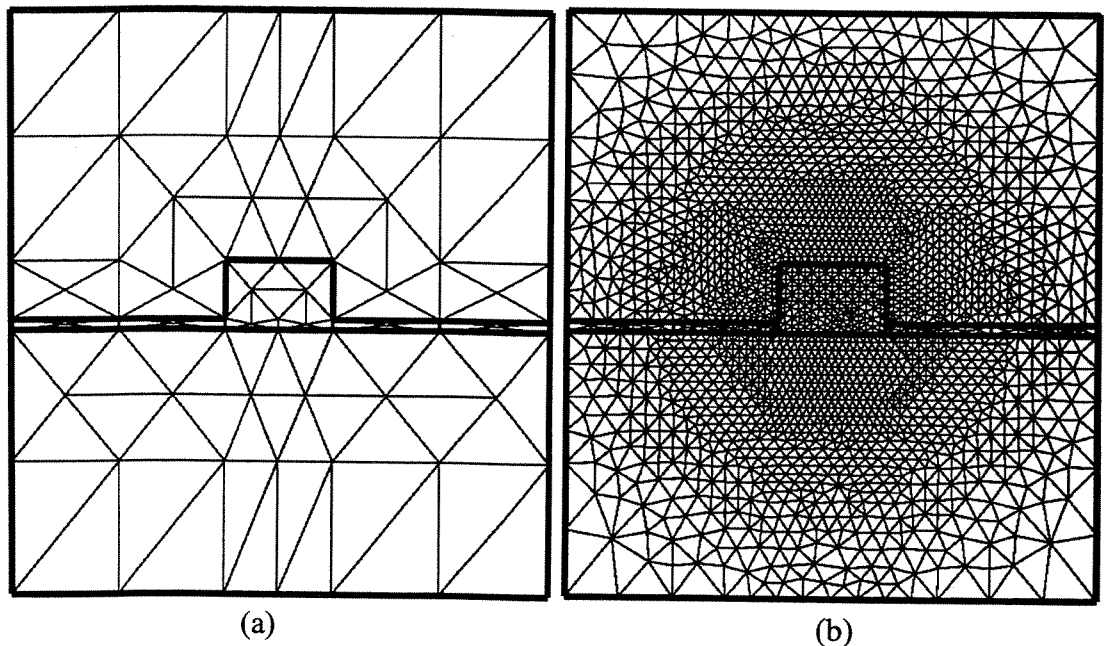


Figura 6.21 – (a) Malha inicial (116 elementos e 71 nós) e (b) malha da solução mais exata (4322 elementos e 2192 nós).

Após 7 iterações, partindo da malha inicial da Figura 6.21(a), chegamos à malha final mostrada na Figura 6.22(a). É possível notar que os elementos se concentram em maior número na região central do guia, conforme era esperado, pois esta é a região onde o campo elétrico se concentra. Na Figura 6.22(b) vemos um detalhe desta região. O valor final obtido para o erro global relativo foi de 0,04%, enquanto que o erro admissível final ficou 0,15% do erro admissível inicial.

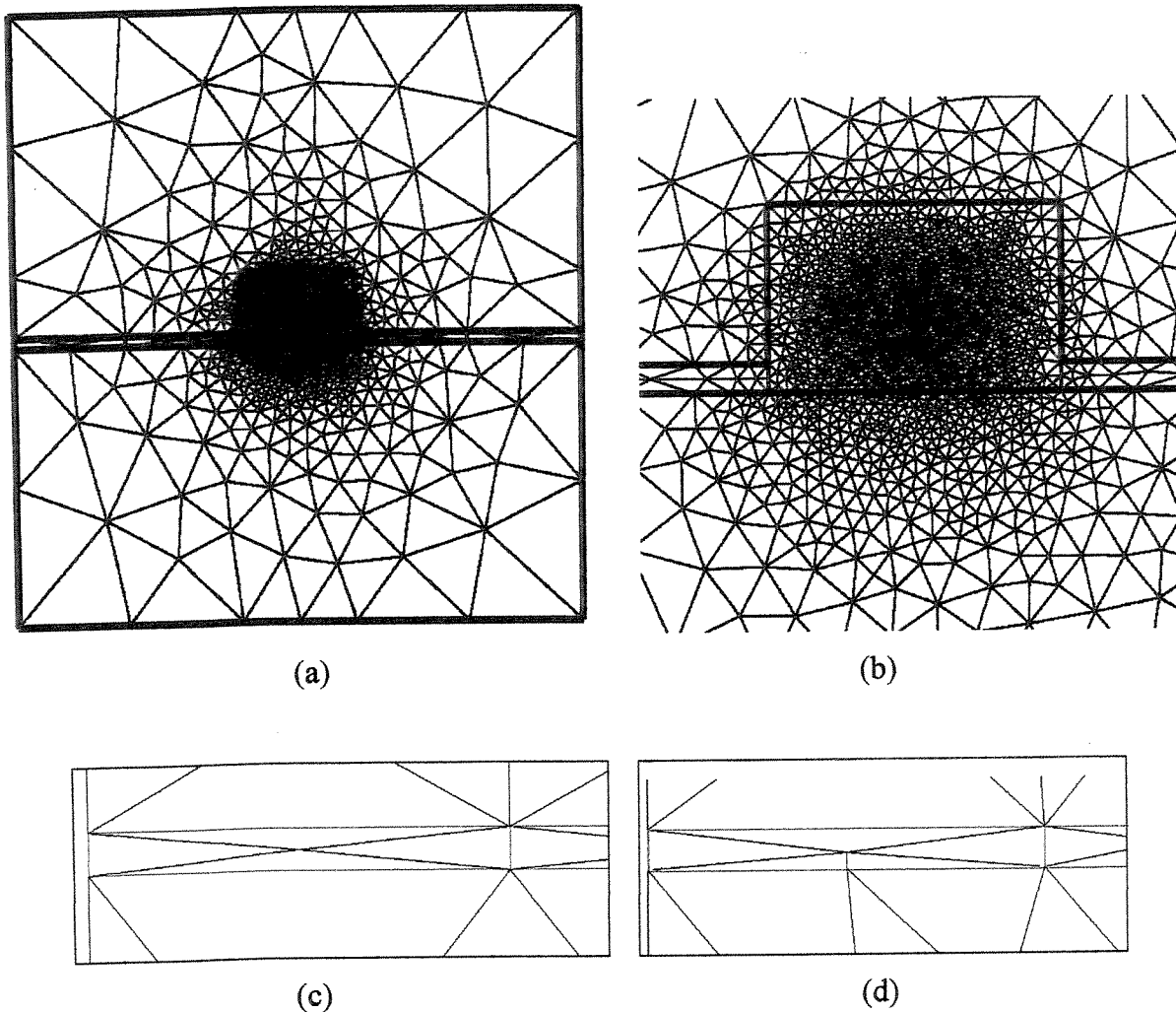


Figura 6.22 – (a) Malha final com 3462 elementos e 1744 nós. (b) Detalhe da região central. Nas Figuras 6.22(c) e (d) mostramos detalhes da região lateral central da malha inicial, Figura 6.21(a), e da malha final, Figura 6.22(a), respectivamente.

Com relação à qualidade final da malha, podemos concluir, a partir do gráfico da Figura 6.23(a), que 93% dos elementos possuem qualidade acima de 0,8. Apesar disso, devido à pequena região mostrada no detalhe na Figura 6.22(c) estar entre duas interfaces, não é possível aplicar a troca de diagonais de forma satisfatória, e por isso os elementos nesta região mantêm seu aspecto ruim na malha final (ver detalhe na Figura 6.22(d)), e o ângulo mínimo permanece quase inalterado, mudando de 5,71 graus na malha inicial para 4,57 graus na malha final. Apesar disso, estes elementos não afetam a qualidade do resultado, uma vez que estão muito distantes da região onde o campo elétrico é mais intenso. Outro fator que atesta a boa qualidade da malha e do resultado numérico, é a grande quantidade de ângulos internos próximos ao ideal (60 graus), como vemos no gráfico da Figura 6.23(b).

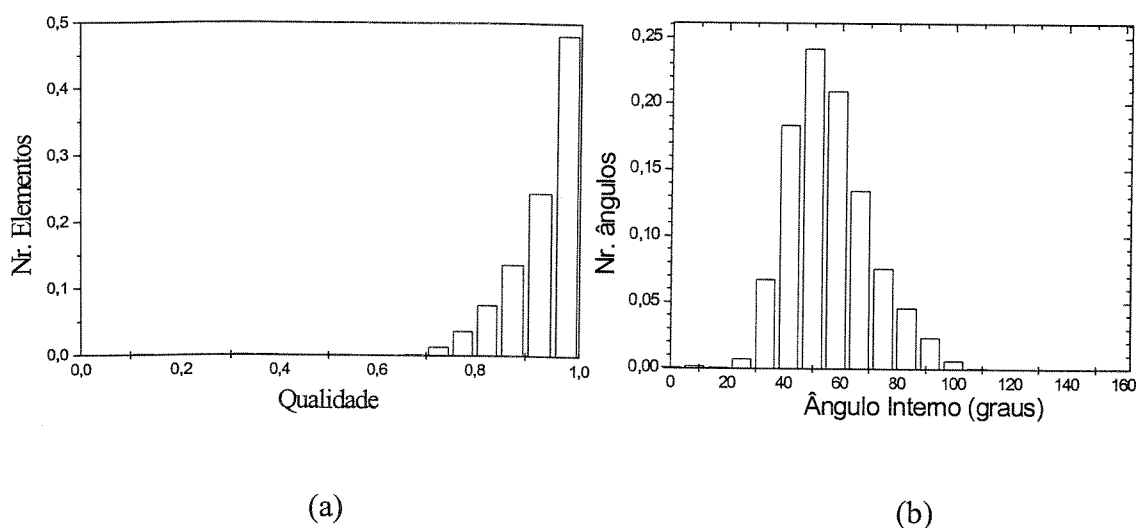


Figura 6.23 – (a) Qualidade e (b) Distribuição de ângulos internos da malha mostrada na Figura 6.22(a).

Após 5 iterações, com uma malha de aproximadamente 1000 elementos, os critérios de parada já estavam satisfeitos; porém, desejávamos estudar também a convergência do valor do índice efetivo obtido com nosso esquema ao valor obtido na literatura [Tsuji98] em função do número de nós. Este estudo é útil para verificarmos a convergência e exatidão de nosso algoritmo. Os resultados podem ser vistos na Figura 6.24.

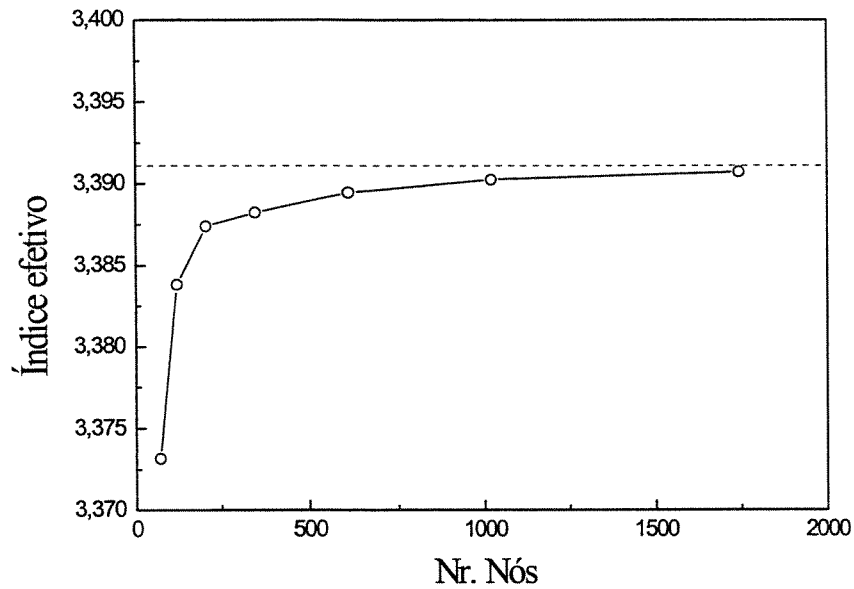


Figura 6.24 – Gráfico da convergência do índice efetivo.

Após analisarmos o gráfico da Figura 6.24, vemos que a nossa solução converge para a melhor solução obtida por Tsuji, dada pela linha tracejada reta, embora ela não atinja o melhor valor de índice efetivo dado pela referência. Esta pequena diferença (ao redor de 0,1%) pode ser explicada se lembrarmos que utilizamos elementos finitos nodais de primeira ordem (lineares) na resolução do problema, enquanto que a solução obtida por Tsuji utilizava elementos finitos de segunda ordem na sua resolução.

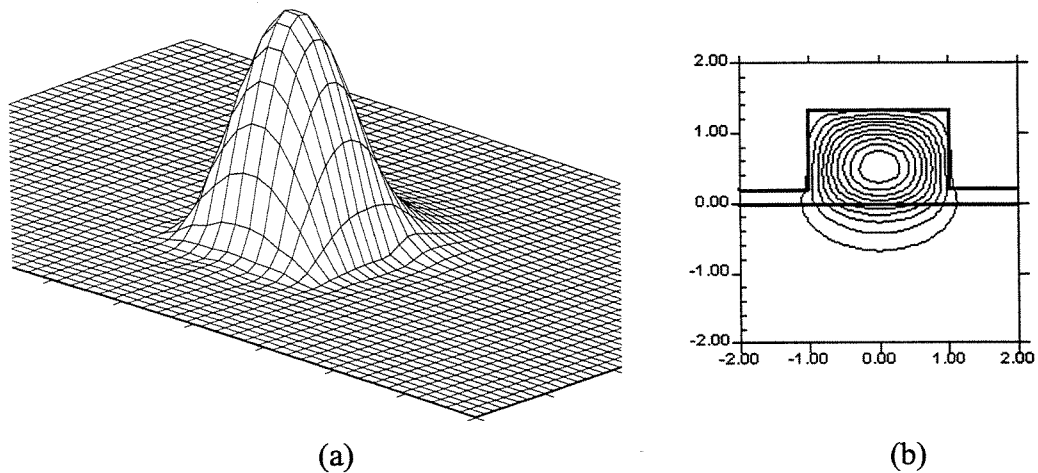


Figura 6.25 – Campo elétrico tridimensional e curvas de contorno.

Nas Figuras 6.25, vemos o campo tridimensional obtido com nossa malha e um traçado das curvas de contorno do campo elétrico sobre o guia de onda costela ('rib'). Se compararmos as Figuras 6.22(b) e 6.25(b), veremos que as regiões de maior refinamento correspondem exatamente às regiões de maior concentração de campo elétrico.

Como podemos concluir, o algoritmo funcionou de forma satisfatória, dentro dos limites de erro esperados, proporcionando a geração adaptativa de uma malha de boa qualidade.

6.2.3.2 – Análise de um Guia de Onda 'Strip' Não-Linear

Nesta Seção analisaremos um guia de onda não-linear composto de um canal, de material linear com índice de refração igual a n_1 , de largura a e altura b , depositado sobre um substrato não-linear, de índice n_3 , e cercado por uma cobertura composta de material linear de índice n_2 , conforme Figura 6.26.

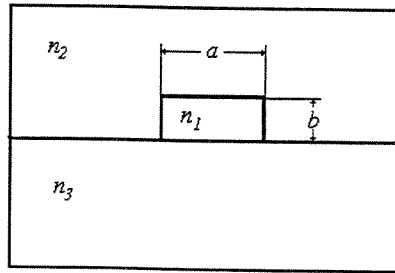


Figura 6.26 – Estrutura do guia de onda não-linear.

A equação para o cálculo através de elementos finitos foi discutida na Seção 5.2.1, e na resolução nós utilizamos elementos finitos nodais lineares, com a condição de guiamento fraco. Como vimos, o índice de refração do substrato terá um fator não-linear, que, por conveniência, iremos repetir aqui:

$$f(\alpha E^2) = 2n_3 \Delta n_{sat} \left[1 - \exp\left(-\frac{\alpha E^2}{2n_3 \Delta n_{sat}}\right) \right] \quad (5.11)$$

com α sendo dado por $\alpha = n_3^2 n_K c \varepsilon_0$, onde n_3 é o índice de refração do substrato, n_K é o coeficiente não-linear de Kerr do meio, c é a velocidade da luz no vácuo e ε_0 é a permissividade elétrica, com Δn_{sat} sendo o máximo aumento não-linear. O efeito esperado da não-linearidade é o aumento do índice do substrato à medida que aumenta a potência no guia de onda, produzindo uma migração do campo elétrico para o substrato.

Para podermos comparar nosso resultado com outros obtidos anteriormente, repetimos os valores numéricos da simulação feita por Li [Li92]. A largura a do canal foi fixada em $2,0 \mu\text{m}$, e a altura em $1,2 \mu\text{m}$. Os valores dos índices de refração são: $n_1 = 1.57$, $n_2 = 1.55$ e $n_3 = 1.55$. O comprimento de onda utilizado foi de $0,515 \mu\text{m}$. Os valores utilizados no cálculo do componente não-linear do índice de refração do substrato foram: $n_K = 10^{-9} \text{ m}^2/\text{W}$ e $\Delta n_{sat} = 0,1$.

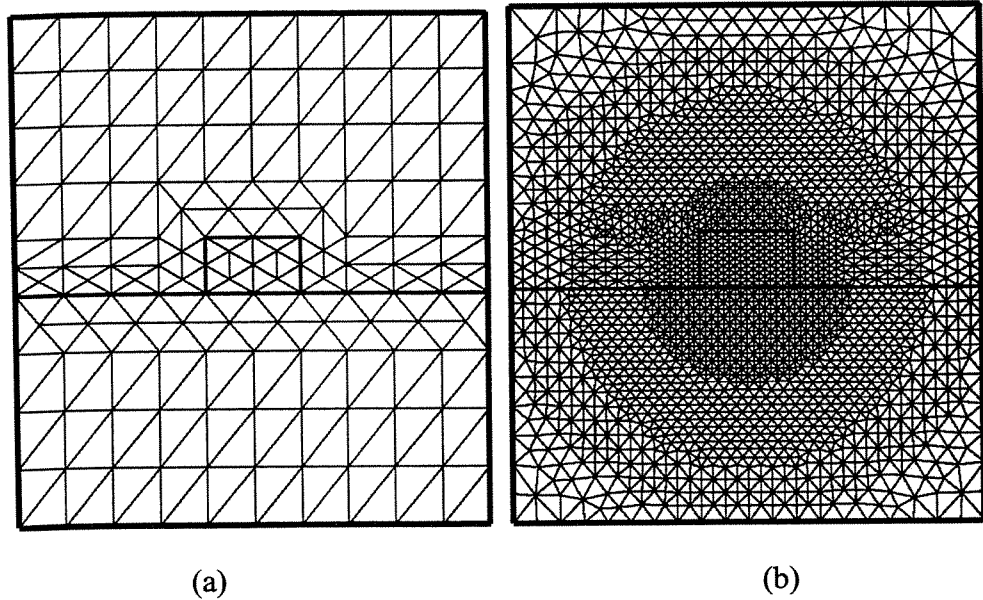


Figura 6.27 – (a) Malha inicial e (b) malha de controle.

Neste exemplo também utilizaremos o método das estimativas de erro para monitorar o refinamento. Os critérios de parada estabelecidos foram o erro global relativo e o erro admissível final. Para o primeiro, esperamos valores menores que 0,5%, enquanto que os valores para o erro admissível final devem ser em torno de 1% do valor inicial.

Como o campo elétrico irá se mover do núcleo para o substrato durante a simulação, a nossa malha também deve acompanhar este movimento.

A malha utilizada para o cálculo da solução de maior aproximação e que será utilizada como solução exata para compararmos com nossos cálculos é mostrada na Figura 6.27(b), e possui 4550 elementos e 2316 nós. Ela foi gerada a partir de uma malha com 256 elementos e 149 nós, mostrada na Figura 6.27(a).

Sabendo que o campo elétrico inicial está mais confinado ao núcleo, tendo um comportamento quase linear devido à baixa variação do índice do substrato, é de se esperar que a malha para o início dos cálculos também esteja mais densa nesta região. Podemos verificar que isto se comprova na Figura 6.28(a), que mostra a malha obtida para o cálculo do campo inicial, com 1112 elementos e 577 nós. Esta malha foi obtida após 4 iterações, e o erro relativo global obtido foi de 0,2 %, com um erro admissível final equivalente a 1,7% do valor inicial. Na Figura 6.28(b) vemos um detalhe da região central.

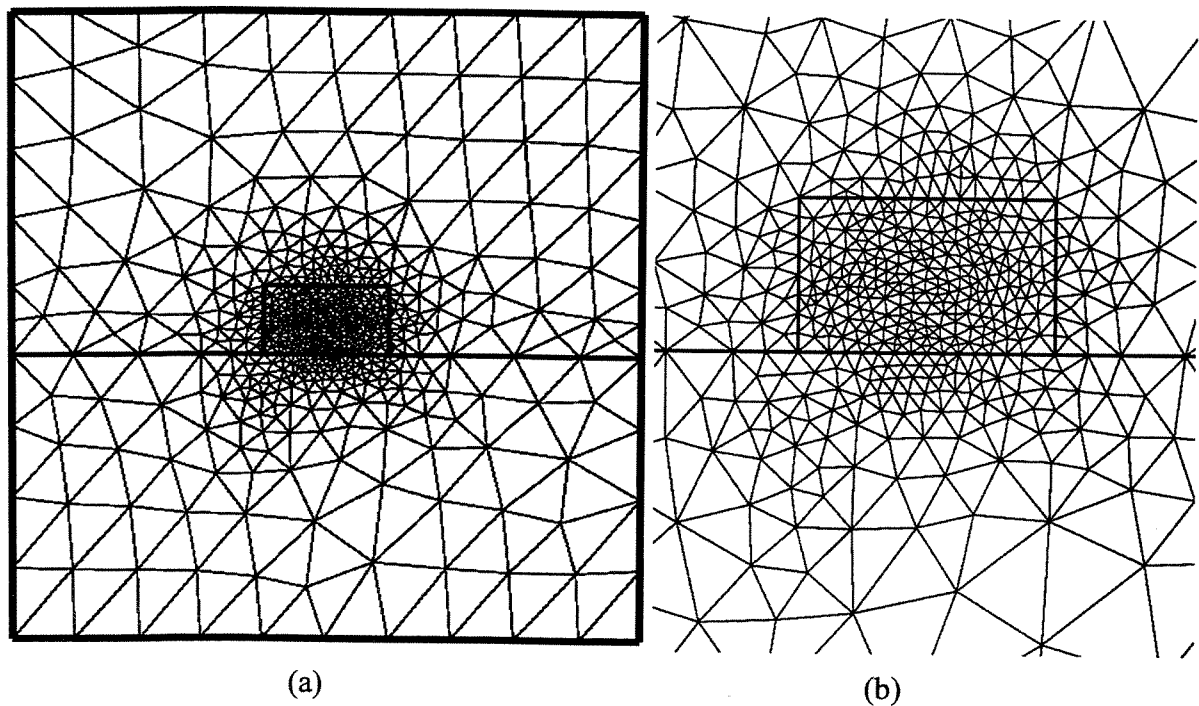


Figura 6.28 – Malha utilizada para (a) o cálculo do campo inicial e (b) detalhe da região central.

Ao final de nosso cálculo, a alta variação no índice de refração do substrato irá deslocar o campo elétrico na sua direção. Portanto, a nossa malha final deverá acompanhar este deslocamento, se concentrando nas regiões de maior densidade de campo. Isto é obtido monitorando-se os erros individuais de cada elemento e refinando aqueles acima do erro admissível, conforme vimos na Seção 2.2. Novamente podemos verificar isto na Figura 6.29(a), onde o refinamento está concentrado em uma região do substrato logo abaixo do núcleo. Esta malha foi obtida após 4 iterações, possuindo 958 elementos e 500 nós. Os valores para a percentagem do erro relativo global e erro admissível final (em relação ao erro inicial) foram de 0,14 % e 0,85 % respectivamente. Na Figura 6.29(b), vemos um detalhe da região de maior refinamento. A malha de controle e a malha inicial foram as mesmas utilizadas para o cálculo do campo inicial, Figuras 6.27.

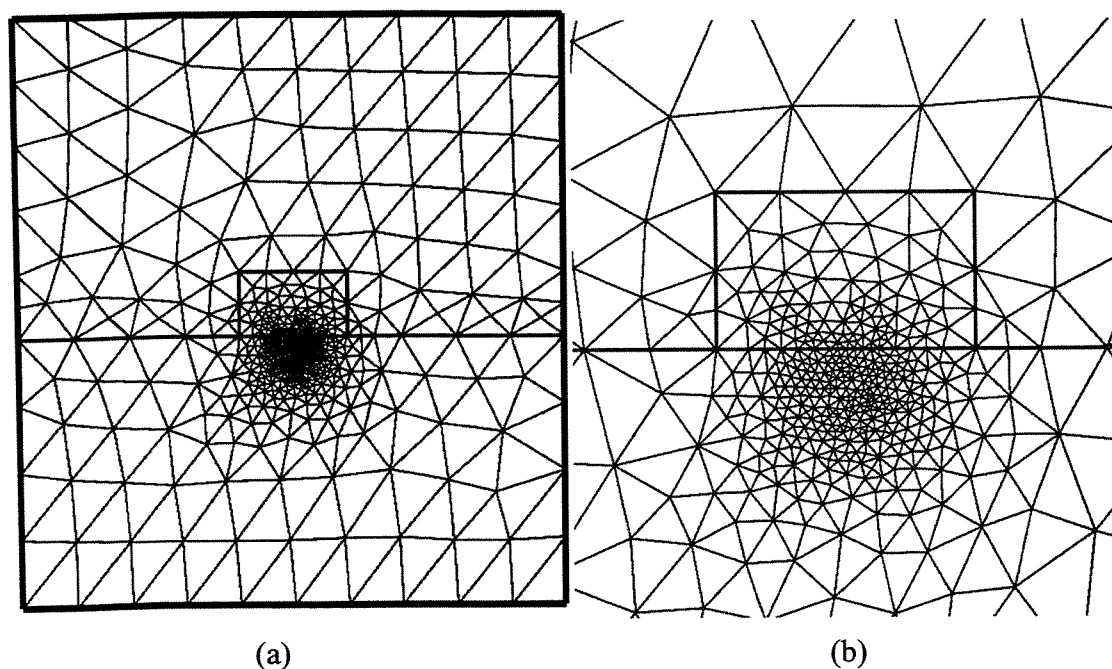


Figura 6.29 – Malha utilizada para (a) o cálculo do campo final e (b) detalhe da região central.

Com relação à qualidade das malhas obtidas, no gráfico da Figura 6.30(a) vemos a distribuição da qualidade da malha final utilizada no cálculo do campo inicial, mostrada na

Figura 6.28, enquanto que na Figura 6.30(b) mostramos a qualidade da malha final, apresentada na Figura 6.29.

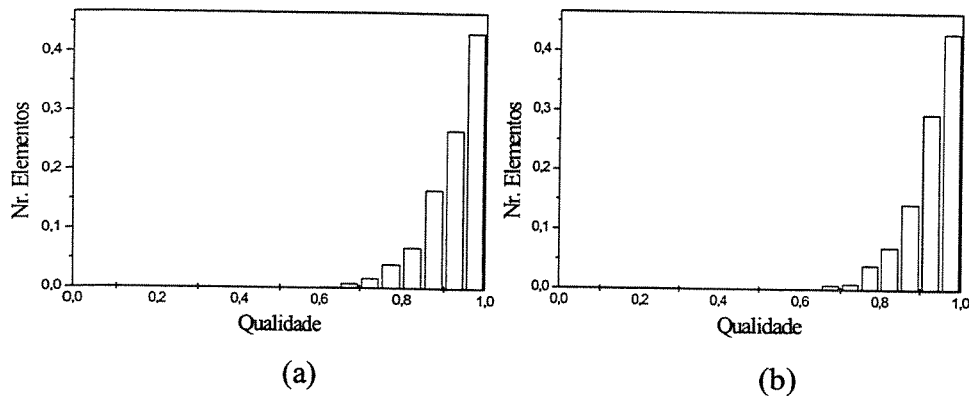


Figura 6.30 – Gráficos da distribuição de qualidade em relação ao número normalizado de elementos das malhas mostradas nas Figuras 6.28 e 6.29, respectivamente.

Podemos ver que, embora o aspecto da malha seja diferente entre ambas, a sua distribuição de qualidade é muito semelhante. Em ambos os casos, em torno de 93% dos elementos possuem qualidade superior a 0,8. Da mesma forma, o ângulo mínimo também se manteve praticamente igual nos dois casos, ficando em torno de 25 graus.

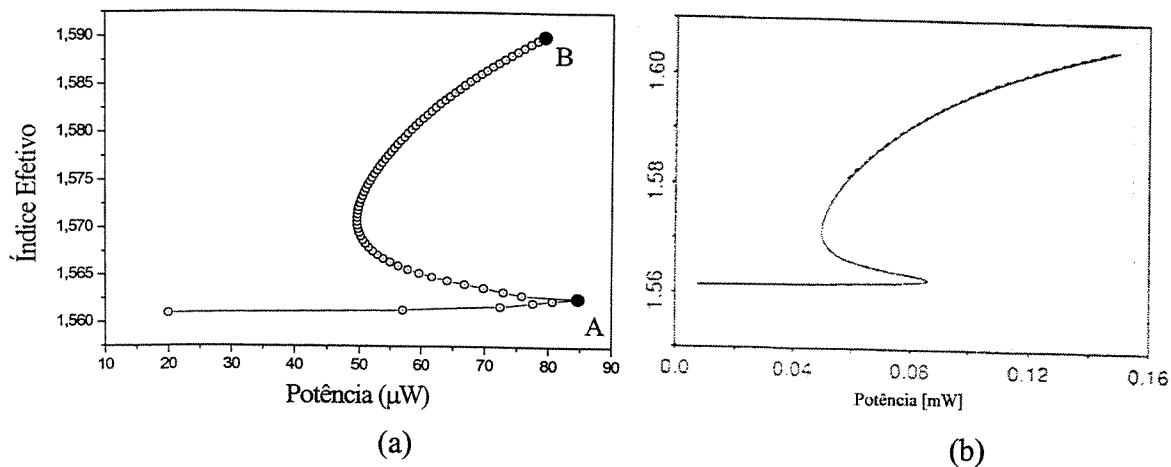


Figura 6.31 – Gráficos da variação do índice efetivo em função da potência. (a) Resultado obtido por nós e (b) por Li.

Após os cálculos, traçamos o gráfico da potência versus índice efetivo, mostrados na Figura 6.30(a), para analisarmos a variação do índice efetivo à medida que elevamos a potência. Comparando os nossos resultados, com aqueles apresentados por Li, na Figura 6.31(b), vemos que, além da forma da curva ser a mesma, o ponto de inflexão *A*, por exemplo, está localizado na mesma posição nos dois gráficos. Se compararmos os gráficos mais detalhadamente, veremos que a aproximação entre ambos é bastante grande, senão exata.

Os pontos *A* e *B* mostram os pontos onde foram calculados os campos inicial e final, respectivamente. Nas Figuras 6.32 mostramos as distribuições de campo elétrico inicial, Figura 6.32(a), e de campo elétrico final, Figura 6.32(b). Nelas é possível ver que o campo inicial está distribuído de uma forma menos concentrada que o campo final.

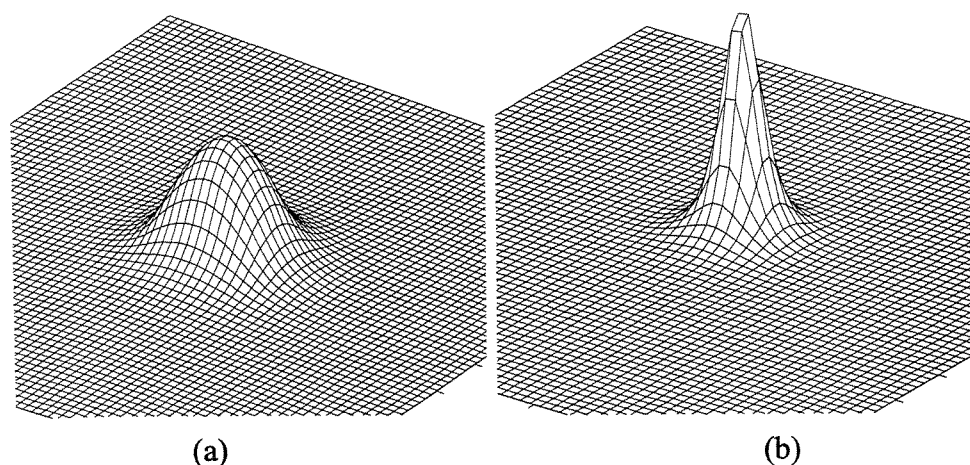


Figura 6.32 – Distribuições de campo obtidas utilizando as malhas mostradas nas Figuras 6.28 e 6.29, respectivamente.

Nos esquemas mostrados na Figura 6.33 vemos as curvas de contorno de campo, aonde fica bem evidente o efeito produzido pelo aumento do índice de refração do substrato não-linear. Na Figura 6.33(a) vemos as curvas de contorno referentes ao campo inicial, mostrado na Figura 6.32(a) e obtidas com a malha mostrada na Figura 6.28. Podemos ver, claramente, que o campo está concentrado no centro do guia, que se comporta de maneira

análoga a um guia tipo ‘rib’ linear. Na Figura 6.33(b), vemos as curvas equipotenciais do campo elétrico final mostrado na Figura 6.32(b) e obtidas com a malha mostrada na Figura 6.29. Neste ponto é possível ver, claramente, a influência do aumento no valor do índice de refração do substrato, decorrente do aumento na potência, causando uma migração do campo em direção ao substrato não-linear. Esta migração ocorre sem a ocorrência de descontinuidades, devido ao fato da estrutura ser fracamente guiada.

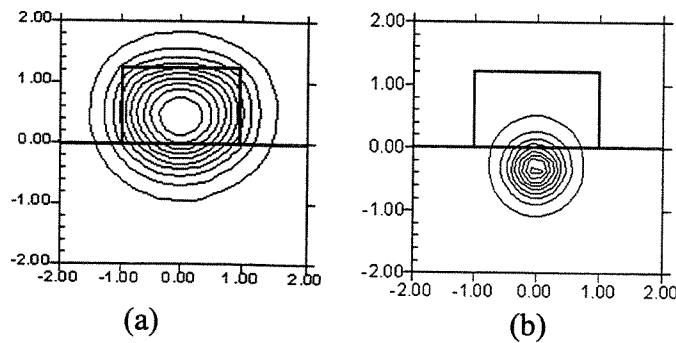


Figura 6.33 – Curvas equipotenciais para os campos (a) inicial e (b) final.

Como resultado final, podemos dizer que após compararmos estes resultados com aqueles já apresentados à comunidade científica, vemos que nosso algoritmo produz bons resultados, tanto no refinamento da malha quanto na resolução numérica do problema.

6.3 – Resultados do Refinamento Tridimensional Superficial

Nesta Seção apresentaremos alguns resultados obtidos utilizando uma adaptação do algoritmo de refinamento bidimensional ao refinamento de superfícies tridimensionais. Nesta Seção, primeiramente mostraremos alguns exemplos de malhas que podemos obter com o auxílio de nossa técnica, analisando-as com respeito à qualidade geométrica dos triângulos, e o tempo de computação.

A seguir, exporemos o resultado de uma simulação numérica obtida com o auxílio de um programa de cálculo através do Método dos Momentos desenvolvido pelo aluno Marcelo I. Davanço, no Departamento de Microondas e Óptica da FEEC/UNICAMP.

6.3.1 – Exemplos de Refinamento

Nesta Seção mostraremos alguns exemplos do refinamento que podemos obter com o auxílio do nosso programa. Estudaremos os resultados para os refinamentos de uma estrutura quadrada simples, uma calota esférica e um cilindro circular. Nestes casos as dimensões das estruturas não têm importância pois queremos analisar a qualidade geométrica das malhas e dos triângulos obtidos.

Como primeiro exemplo, mostraremos o refinamento de uma estrutura quadrada simples. Na Figura 6.34(a) vemos a malha inicial com 6 elementos e 7 nós. O ângulo interno mínimo é de 45 graus, enquanto que o ângulo interno máximo é de 90 graus. Como todos os elementos desta malha são iguais, a qualidade máxima e mínima também é a mesma, sendo de 0,866.

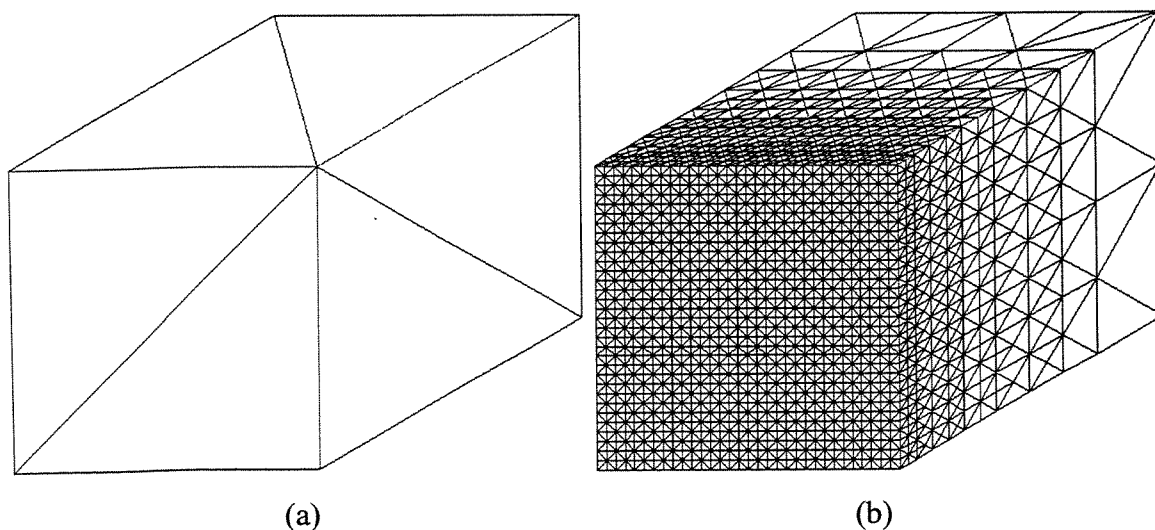


Figura 6.34 – Refinamento superficial de uma estrutura quadrada. (a) Malha inicial e (b) malha final.

Na Figura 6.34(b) temos a malha final obtida com um refinamento centrado na face frontal. O número de elementos final foi de 3128 enquanto que o número total de nós chegou a 1611.

Como todos os elementos iniciais são triângulos retângulos, a constante subdivisão dos mesmos irá gerar novos triângulos retângulos de uma forma similar à encontrada na geometria fractal, conforme previsto na literatura [Rivara96]. Como não é feita a relaxação dos nós, os elementos resultantes finais tem o mesmo aspecto dos elementos iniciais, que são todos iguais. Com isso a qualidade final também será a mesma dos triângulos iniciais, que é de 0,866. Podemos verificar isso na distribuição de qualidade em função do número normalizado de elementos, Figura 6.35(a). Se analisarmos a distribuição de ângulos internos, Figura 6.35(b), também veremos que 2/3 deles são de 45 graus e 1/3 de 90 graus, mostrando a geração de triângulos retângulos semelhantes aos originais.

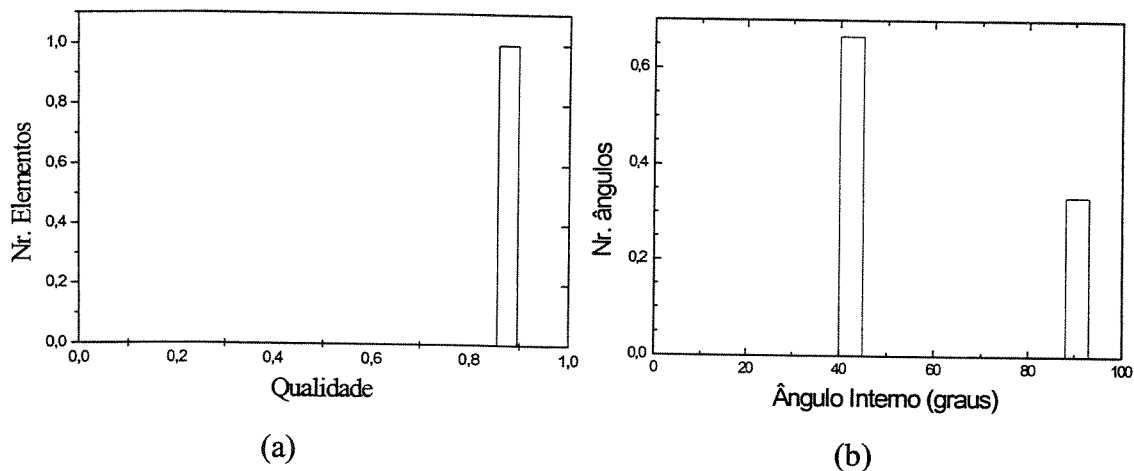


Figura 6.35 – Distribuição (a) da qualidade e (b) dos ângulos internos da malha mostrada na Figura 6.34(b).

No próximo exemplo, mostraremos o refinamento de uma calota esférica, ou meia esfera. Na Figura 6.36(a) vemos a malha inicial com 4 elementos e 5 nós. Estes elementos iniciais são todos triângulos equiláteros, portanto a sua qualidade será igual a 1,0, e os ângulos internos serão iguais a 60 graus. Na Figura 6.36(b) temos uma vista lateral da

malha final, que contém 5062 elementos e 2545 nós. Na Figura 6.36(c) temos uma vista superior da mesma malha. As funções de densidade foram escolhidas de forma a monitorar um refinamento maior na parte superior da calota.

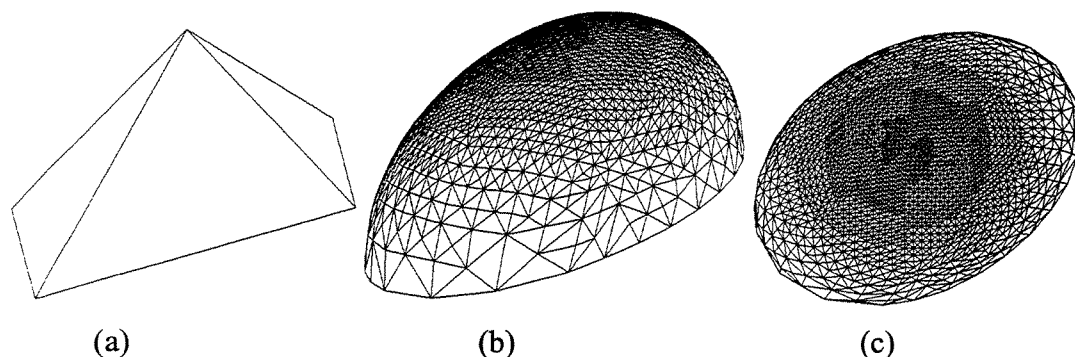


Figura 6.36 – Refinamento de uma calota esférica. (a) Malha inicial, (b) Visão lateral da malha final e (c) Visão superior da malha final.

Analisando a malha final com respeito à qualidade da mesma, concluímos que a qualidade mínima obtida foi de 0,672 e a máxima foi de 0,982, com 60 % dos elementos com qualidade superior a 0,8, conforme Figura 6.37(a). Como vimos na Seção 3.4, a troca de diagonais em superfícies tridimensionais pode acarretar algum erro, mas analisando a malha final vemos que este erro é pequeno e não irá comprometer o resultado final de simulações numéricas.

Quanto à distribuição de ângulos internos, vemos que existe uma grande quantidade de elementos com ângulos ao redor de 45 graus, com picos menores ao redor de 65, 95 e 105 graus. O ângulo mínimo foi de 28,1 graus e ângulo máximo foi de 113,2 graus.

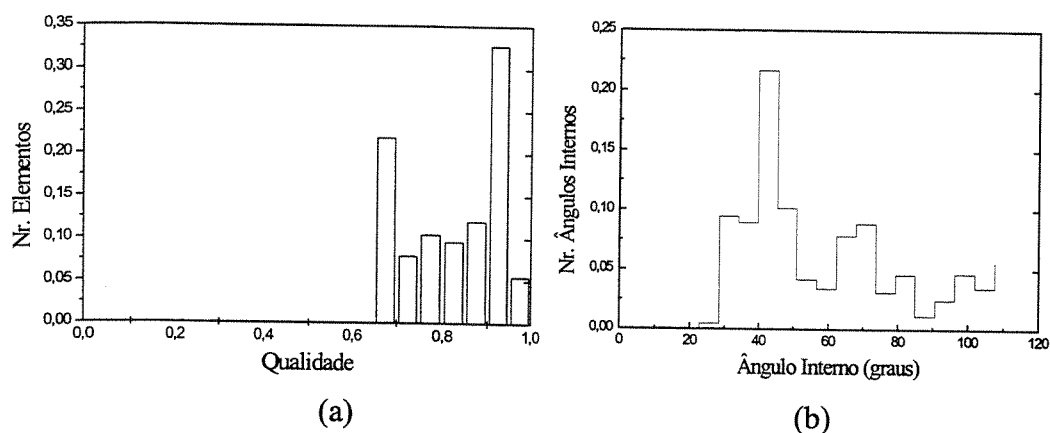


Figura 6.37 – Distribuição (a) da qualidade e (b) dos ângulos internos da malha final.

O último exemplo nos mostra o refinamento de um cilindro de raio constante. Devido à simetria do mesmo, utilizamos apenas metade da estrutura no refinamento. Na Figura 6.38(a) vemos a malha inicial com 8 elementos e 8 nós. O ângulo mínimo obtido foi de 42,56 graus e o máximo chegou a 94,86 graus, enquanto que a qualidade mínima e máxima foram de 0,82 e 0,94, respectivamente. Na Figura 6.38(b) vemos a malha final com 3248 elementos e 1665 nós. O ângulo mínimo e máximo finais ficaram em 43,25 e 93,48 graus respectivamente, enquanto que a qualidade da malha variou de 0,83 a 0,89. As funções de densidade foram escolhidas de forma a produzir um refinamento maior na parte frontal da malha.

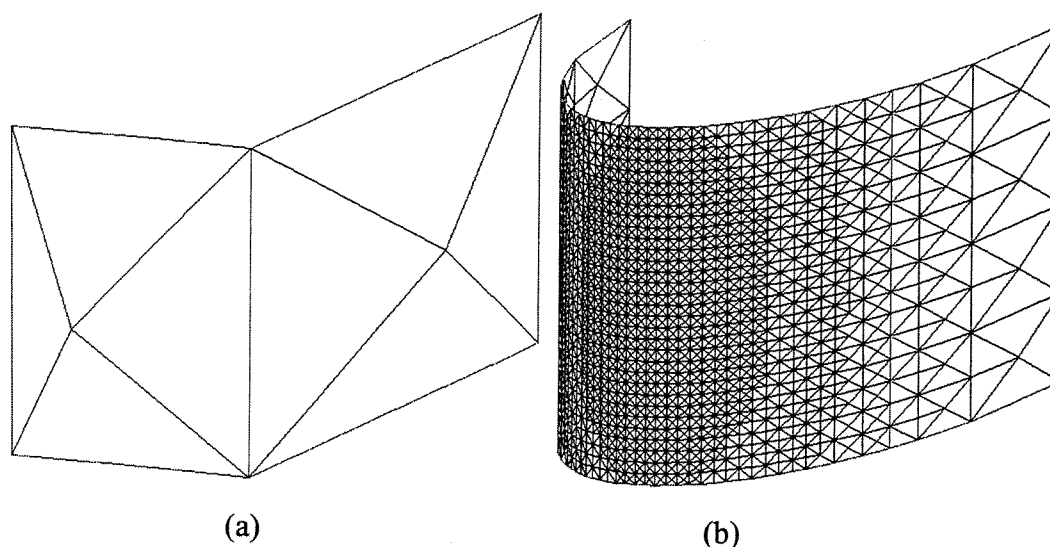


Figura 6.38 – Refinamento de um cilindro, (a) Malha inicial e (b) Malha final.

Analisando quanto à qualidade, é possível vermos que esta malha é de qualidade muito boa, pois todos os elementos possuem qualidade superior a 0,8, conforme Figura 6.39(a). De acordo com a Figura 6.39(b), vemos dois picos, ao redor de 45 e 90 graus, o que nos sugere um grande número de triângulos retângulos. O uso da troca de diagonais, neste caso, aumentou um pouco o valor do ângulo mínimo, e diminuiu o valor do ângulo máximo, aumentando também o valor mínimo da qualidade, embora a qualidade máxima tenha diminuído.

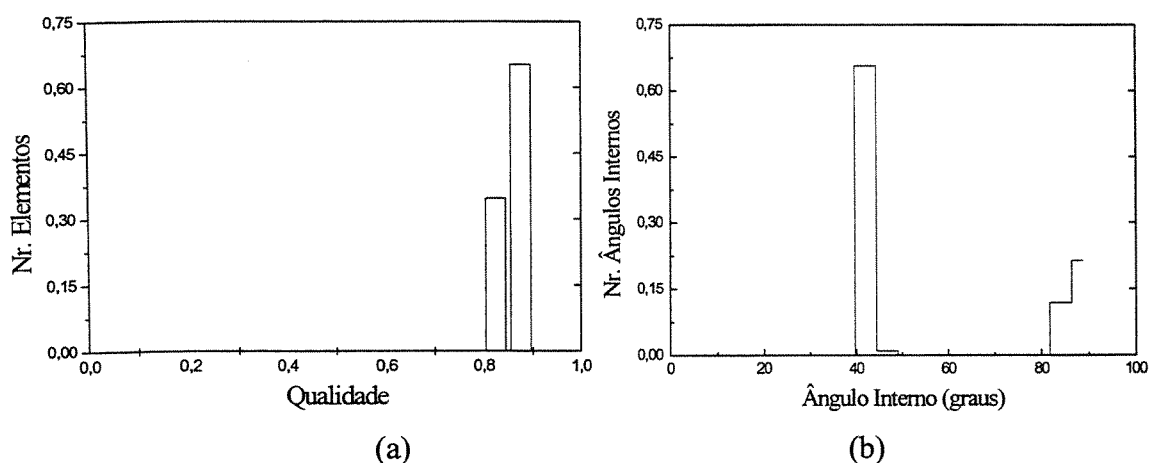


Figura 6.39 – Distribuição (a) da qualidade e (b) dos ângulos internos para a malha mostrada na Figura 6.38(b).

Analisando os resultados dos exemplos estudados nesta Seção, vemos que o nosso algoritmo de refinamento é capaz de produzir malhas de boa qualidade a partir de malhas iniciais de aspecto bastante simples. O algoritmo também é capaz de adaptar a malha a superfícies de formato pré-definido, tais como esferas, cilindros, etc.

Os tempos de refinamento para a obtenção destas malhas é semelhante ao obtido no refinamento de malhas bidimensionais.

6.3.2 – Estudo Numérico.

Nesta Seção estudaremos o espalhamento de uma onda eletromagnética por uma esfera condutora de raio unitário. Este estudo tem por finalidade verificar se a malha conseguida com a ajuda de nosso algoritmo produz resultados numéricos satisfatórios.

Para os cálculos, produzimos uma esfera com 386 nós e 768 elementos, com um refinamento uniforme sobre toda a superfície da esfera, que pode ser vista na Figura 6.40. Os ângulos internos dos elementos da malha final variaram de um mínimo de 30,18 graus a um máximo de 100,22 graus, enquanto que a qualidade final dos elementos ficou entre 0,68 e 0,92.

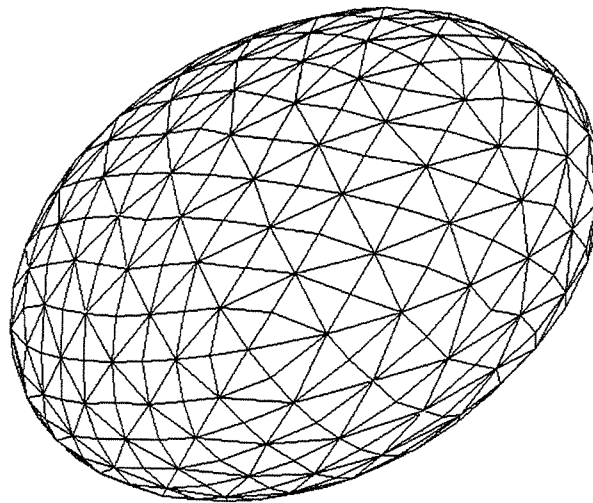


Figura 6.40 – Malha final para o cálculo numérico.

Nas Figuras 6.41 vemos os gráficos de distribuição de qualidade e ângulos internos da malha final. Podemos notar que a qualidade e os ângulos internos da malha final seguem o padrão de boa qualidade obtido anteriormente nos exemplos.

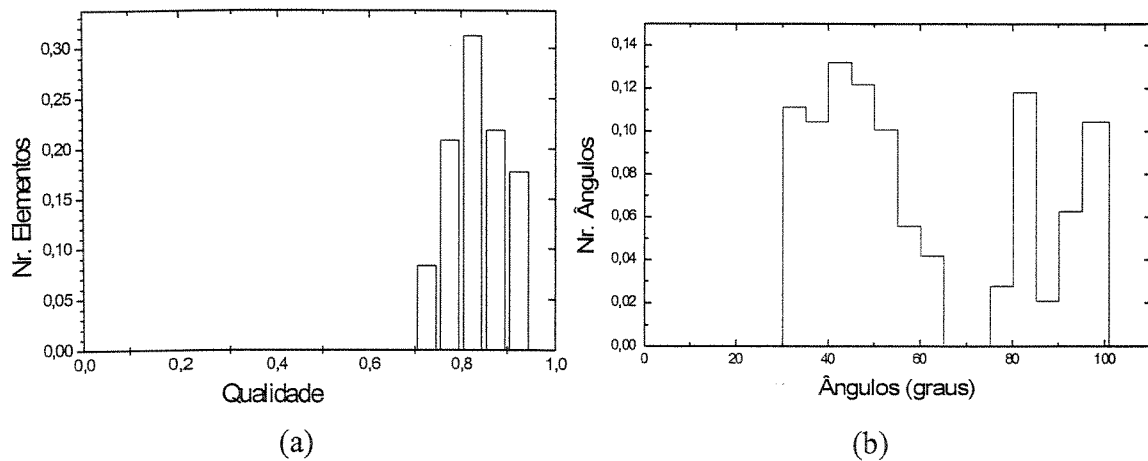


Figura 6.41 – Distribuição (a) da qualidade e (b) dos ângulos internos da malha mostrada na Figura 6.40.

Utilizando o método dos momentos, apresentado na Seção 5.3, resolvemos a Equação 5.23. Com o auxílio desta equação foi possível calcularmos a seção reta de espalhamento bi-estático $\sigma_{\theta\theta}$, dada pelas equações 5.25. Neste exemplo, utilizaremos uma esfera de raio unitário e a constante de propagação da onda, k , será igual a 2,30. Esta onda eletromagnética está incidindo da direção z . Estes valores foram escolhidos a fim de reproduzir os dados apresentados por Wang [Wang91].

Nas Figuras 6.42 e 6.43 é possível vermos a malha esférica e a distribuição superficial normalizada de corrente. Na Figura 6.42 temos a visão lateral da esfera, correspondendo ao plano xz . Como vemos, a distribuição de corrente é maior no topo da esfera, aonde incide a onda, e diminui gradativamente até o fundo da esfera.

Nas Figuras 6.43(a) temos uma visão do topo da esfera e na Figura 6.43(b) mostramos o fundo da mesma.

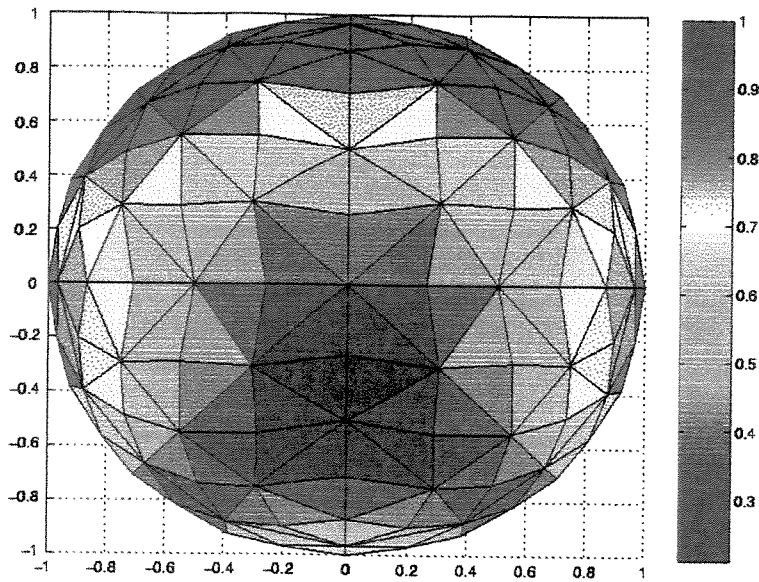


Figura 6.42 – Vista lateral da distribuição superficial normalizada de corrente na esfera.

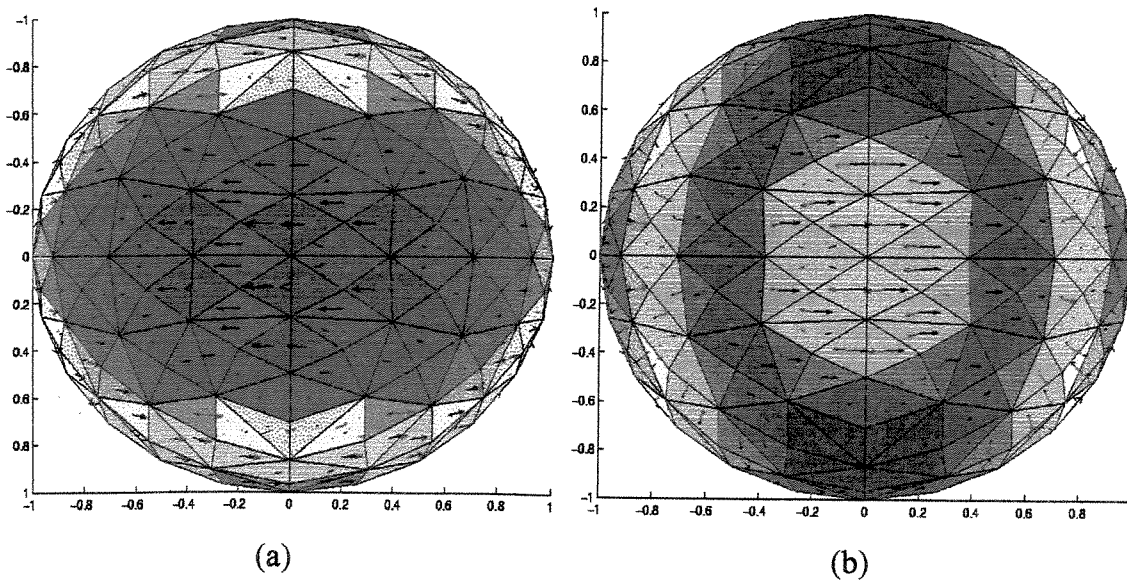


Figura 6.43 – Distribuição superficial normalizada de corrente na esfera: (a) Vista Superior e (b) Vista inferior.

Na Figura 6.44 vemos os gráficos da seção reta de espalhamento bi-estático para uma onda E incidente. Na Figura 6.44(a) apresentamos o resultado obtido com o auxílio da nossa malha, enquanto que na Figura 6.44(b) vemos o gráfico do resultado obtido por

Wang. Fazendo a comparação é possível dizer que a semelhança entre ambos é muito grande. O mesmo podemos dizer se compararmos os nossos resultados para o espalhamento bi-estático de onda **H** incidente, Figura 6.44(c), e os resultados obtidos por Wang, Figura 6.44(d).

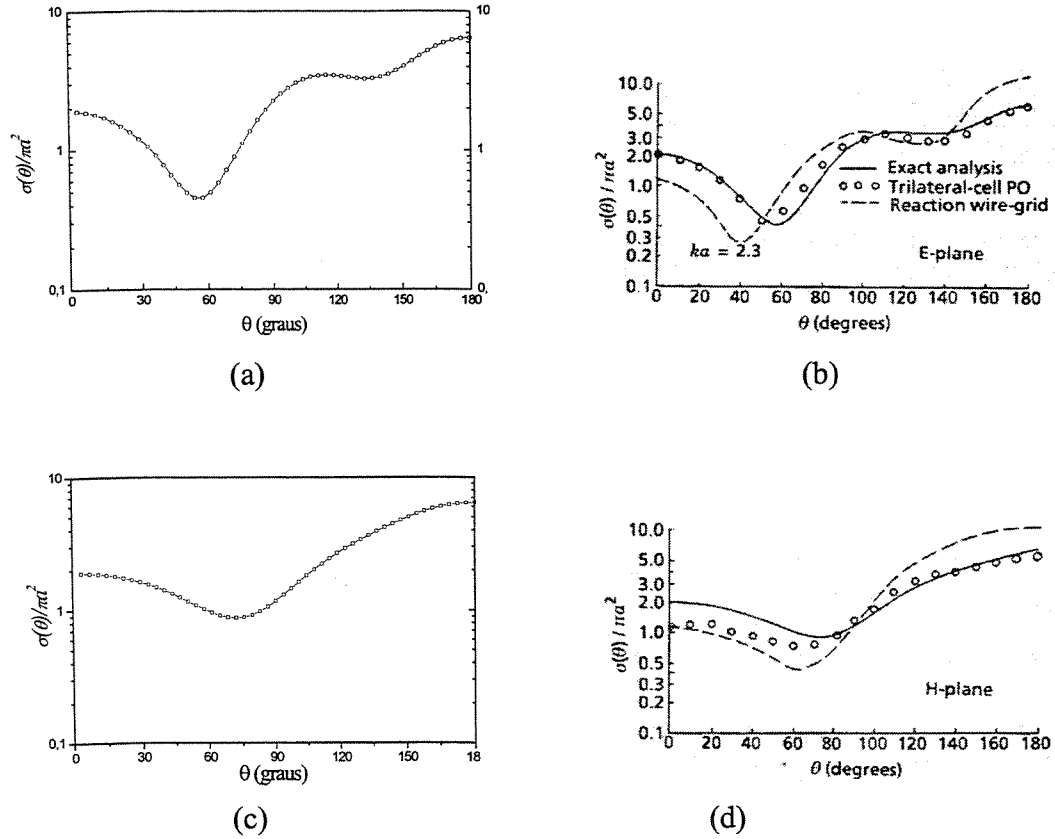


Figura 6.44 – Comparação entre os resultados para o cálculo da seção reta do espalhamento bi-estático de uma onda incidente **E**, Figuras (a) e (b), e **H**, Figuras (c) e (d). Os dados correspondentes às Figuras (a) e (c) foram obtidos por nós, enquanto que os dados das Figura (b) e (d), foram obtidos por Wang [Wang91].

6.4 – Resultados do Refinamento Tridimensional Volumétrico

Finalmente, apresentaremos agora os resultados para o refinamento tridimensional de estruturas volumétricas, aonde passaremos a utilizar tetraedros. Conforme dissemos no Capítulo 6, algumas mudanças foram necessárias para a implementação deste algoritmo.

De forma semelhante ao que fizemos na apresentação dos resultados dos refinamentos bidimensional e tridimensional superficial, primeiramente apresentaremos alguns exemplos de malhas obtidas com o auxílio de nosso programa, analisando-as com respeito às suas qualidades geométricas, e, em seguida, faremos uma simulação numérica .

6.4.1 – Exemplos de Refinamento

Nesta Seção, apresentamos alguns exemplos de refinamento que obtivemos com o auxílio de nosso algoritmo. Estes exemplos foram feitos para estudarmos as qualidades geométricas do refinamento. Infelizmente a visualização da malha fica prejudicada devido às suas características tridimensionais.

Os casos estudados foram o refinamento de um tetraedro e de um cubo. Estudamos ambos os casos pois ambos podem ser usados como primitivas na construção de estruturas mais complexas.

No primeiro caso, partimos de uma malha inicial com apenas 4 elementos e 5 nós, mostrada na Figura 6.45(a). O ângulo mínimo e máximo foram de 33,67 e 110,59 graus, respectivamente, enquanto que a qualidade variou de 0,48 a 0,52. Após o refinamento, chegamos à malha final, mostrada na Figura 6.45(b), que contém 795 elementos e 216 nós. As linhas mais grossas definem a superfície externa da estrutura. O refinamento foi monitorado por funções de densidade de formato gaussiano pré-definido, de forma a preencher a estrutura com mais tetraedros na região direita da estrutura.

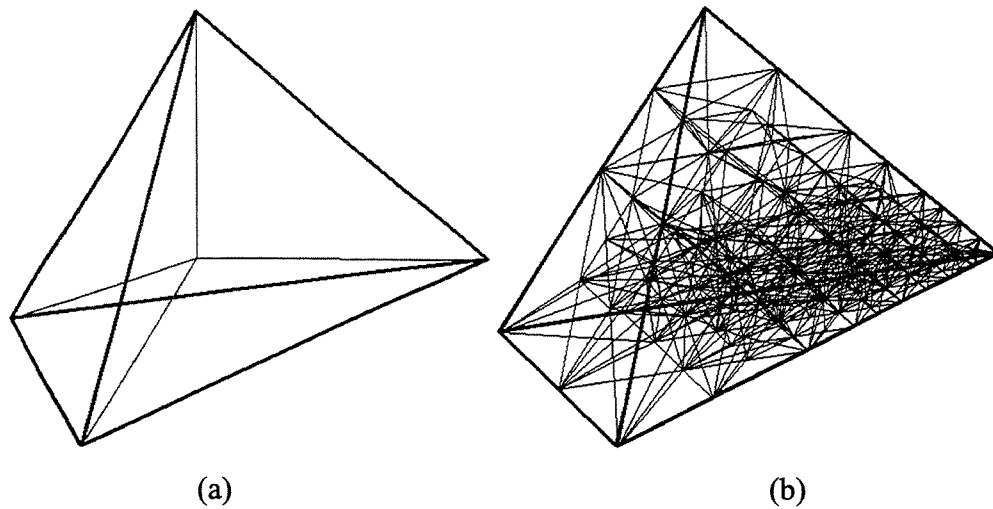


Figura 6.45 – Refinamento de uma estrutura tetraédrica. (a) Malha inicial e (b) malha final.

Quanto às distribuições de qualidade e ângulo interno da malha da Figura 6.45(b), os gráficos correspondentes são mostrados na Figura 6.46. Como podemos ver a qualidade da maioria dos tetraedros não fica próxima à qualidade equivalente a triângulos equiláteros, como acontecia no refinamento de triângulos. Aqui a qualidade de grande parte dos elementos permanece próxima dos valores de qualidade dos tetraedros iniciais, com variações para melhor e para pior. O máximo atingido foi de 0,936 enquanto que a qualidade mínima foi de 0,202.

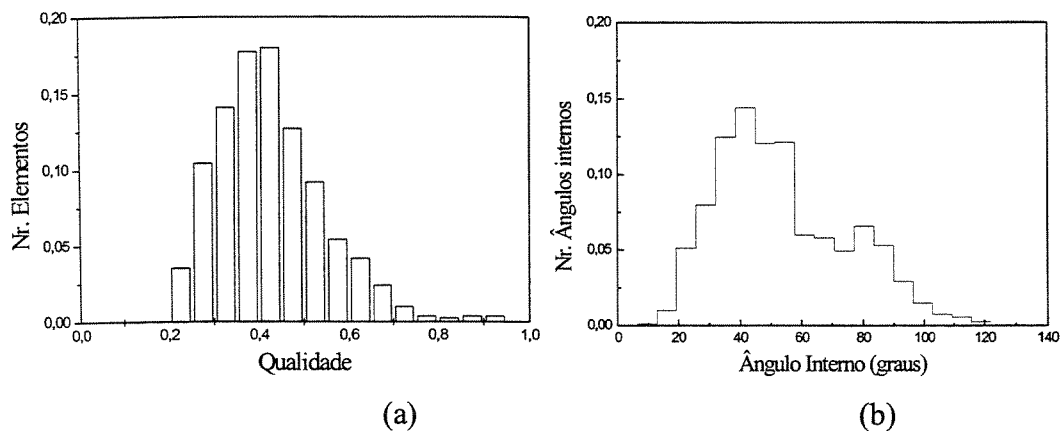


Figura 6.46 – Distribuições (a) da qualidade da malha em relação ao número de elementos e (b) do ângulo interno em relação à quantidade normalizada de ângulos internos.

Quanto à distribuição de ângulos internos, vemos que ela se mantém semelhante a alguns casos obtidos para os refinamentos bidimensionais e superficiais. O ângulo mínimo obtido foi de 17,47 graus, portanto de acordo com a Seção 2.3.2, onde dissemos que o ângulo mínimo final seria maior ou igual a metade do ângulo mínimo inicial (33,67 graus), enquanto que o máximo chegou a 128,44 graus. A partir da Figura 6.46(b), podemos deduzir que os ângulos agudos (menores que 30 graus) chegam a 6% do total.

No segundo exemplo estudamos o refinamento de um cubo. Na Figura 6.47(a) vemos a malha inicial com 12 elementos e 9 nós. O refinamento foi monitorado por funções de formato gaussiano de forma a produzir um malhamento maior na parte posterior da estrutura, o que infelizmente é difícil de se verificar visualmente. A qualidade máxima e mínima dos elementos iniciais são de 0,342 e 0,359, respectivamente. O ângulo mínimo foi de 35,26 graus, enquanto que o máximo chegou a 109,47 graus.

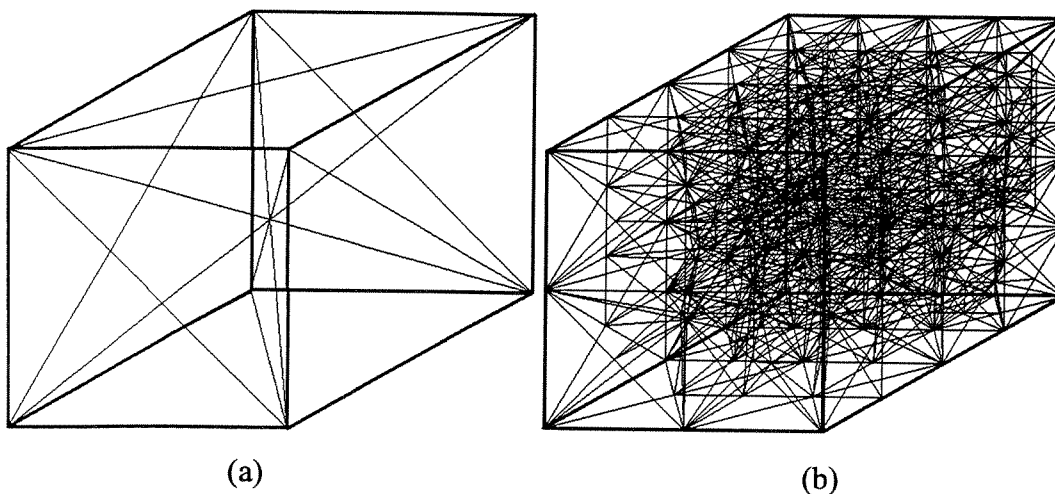


Figura 6.47 – (a) Malha inicial e (b) malha final para o estudo do refinamento de um cubo.

A distribuição de qualidade da malha final é mostrada na Figura 6.48(a), aonde o mínimo é 0,248 e o máximo é 0,689, estando o pico ao redor de 0,4. Apesar de termos elementos com qualidades inferiores à qualidade mínima inicial (21 %), a quantidade de elementos com qualidade superior à qualidade máxima inicial é maior (53 %).

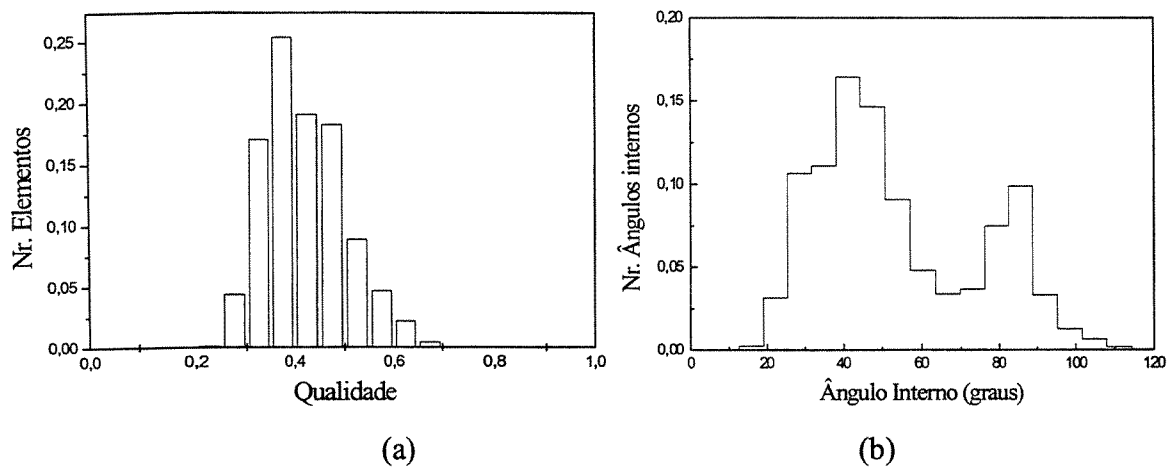


Figura 6.48 – Distribuição (a) da qualidade e (b) dos ângulos internos da malha final.

Com relação à distribuição de ângulos internos, na Figura 6.48(b) vemos que existem dois picos distintos ao redor de 45 e 90 graus. O ângulo máximo obtido foi de 127 graus, enquanto que o ângulo mínimo foi de 21,3 graus, novamente de acordo com o que foi dito na Seção 2.3.2. A quantidade de ângulos inferiores a 30 graus corresponde a 3,2 % do total.

Quanto ao tempo de processamento, infelizmente, devido ao aumento de dados a serem armazenados e manipulados, o tempo de computação também aumentou bastante, como podemos ver no gráfico do tempo de processamento, Figura 6.49.

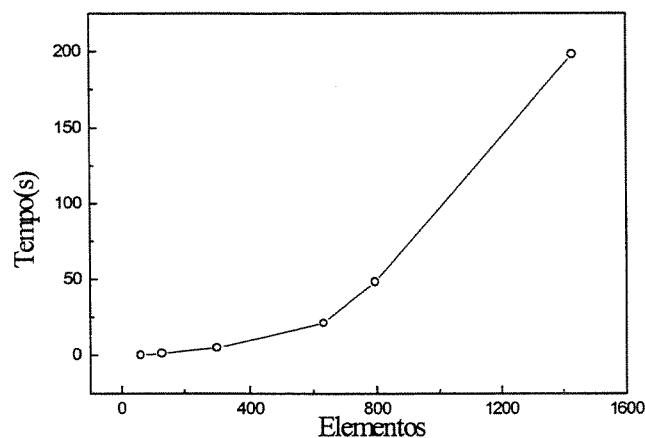


Figura 6.49 – Tempo de processamento versus Quantidade de elementos.

Como vimos nestes exemplos, as malhas obtidas com o auxílio de nosso algoritmo não possuem uma qualidade tão boa quanto àquelas obtidas para os refinamentos bidimensionais e tridimensionais superficiais, com um tempo de execução superior a ambas. Entretanto, conseguimos evitar a formação de tetraedros do tipo ‘lasca’ que conforme vimos na Seção 4.1, são extremamente indesejáveis. Com a implementação futura de técnicas de transformação 2→3, 3→2, 4→4, etc ... , discutidas na Seção 4.2, a qualidade da malha tende a melhorar.

6.4.2 – Exemplo Numérico

Nesta Seção, analisaremos o desempenho de uma malha de elementos tetraédricos produzida com o auxílio de nosso algoritmo em um estudo numérico. Neste estudo computaremos as frequências de ressonância de uma cavidade retangular fechada.

Este problema possui uma solução analítica [Collin66], que nos diz que as frequências de ressonância para uma cavidade retangular fechada e oca são dadas pela equação

$$\omega = c \sqrt{\left(\frac{l \cdot \pi}{d}\right)^2 + \left(\frac{m \cdot \pi}{b}\right)^2 + \left(\frac{n \cdot \pi}{a}\right)^2} \quad (6.1)$$

com ω sendo a frequência angular de ressonância, c é a velocidade da luz no vácuo, $d \times b \times a$ são as dimensões da cavidade e TE_{lmn} é o modo que se deseja calcular. A estrutura da cavidade retangular é mostrada na Figura 6.50. As dimensões utilizadas foram: $d = 1m$, $b = 0,4m$ e $a = 0,3m$.

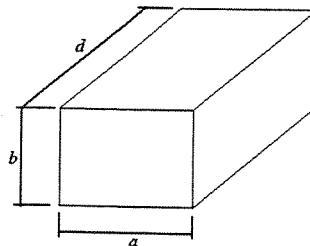


Figura 6.50 – Cavidade retangular e dimensões.

Na Figura 6.51 mostramos a malha inicial, com 12 elementos e 9 nós, e a malha final, com 2589 elementos, 622 nós e 3480 arestas. O refinamento foi monitorado de forma a produzir uma malha uniforme. O ângulo mínimo inicial foi de 15,56 graus e o máximo chegou a 148,8 graus. A qualidade inicial variou entre 0,222 e 0,688.

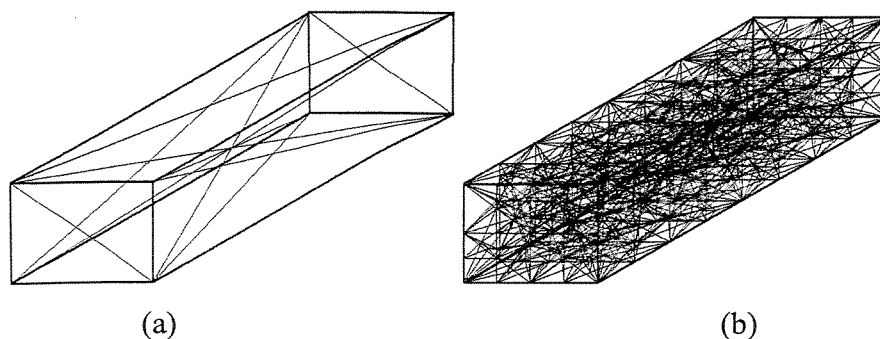


Figura 6.51 – Refinamento de uma cavidade retangular tridimensional. (a) Malha Inicial e (b) Malha Final.

Com relação à qualidade e distribuição de ângulos internos da malha final, analisando os gráficos das Figuras 6.52, vemos que a qualidade se mantém ao redor de um pico na região de 0,4. Os elementos com qualidades inferiores à mínima inicial correspondem a, aproximadamente, 5,5% do total, enquanto que os elementos com qualidade geométrica superior à qualidade máxima inicial somam em torno de 3,5% do total. A qualidade máxima final foi de 0,801, enquanto que a qualidade mínima final chegou a 0,136.

O gráfico dos ângulos internos nos mostra que eles estão distribuídos de forma quase-simétrica em torno de um pico na região de 45 graus, com uma formação bastante acentuada de ângulos com 60 graus. A percentagem de ângulos agudos chega a 13% do total. O ângulo mínimo final obtido foi de 13,45 graus enquanto que o máximo ficou em 146,6 graus.

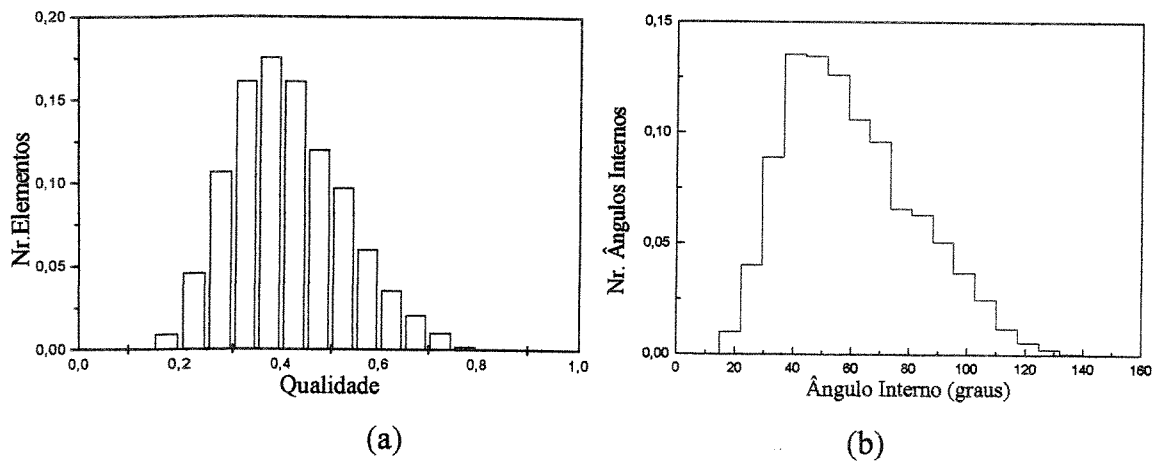


Figura 6.52 – Distribuição (a) da qualidade e (b) dos ângulos internos na malha final.

Pela análise da malha inicial, vemos que a mesma possui alguns ângulos internos muito agudos, além de uma grande variação entre a maior e a menor aresta. Como nosso algoritmo, no atual estágio, só faz a suavização das coordenadas dos elementos internos e a troca de diagonais em elementos que ficam na superfície externa, não foi possível corrigir estas imperfeições iniciais.

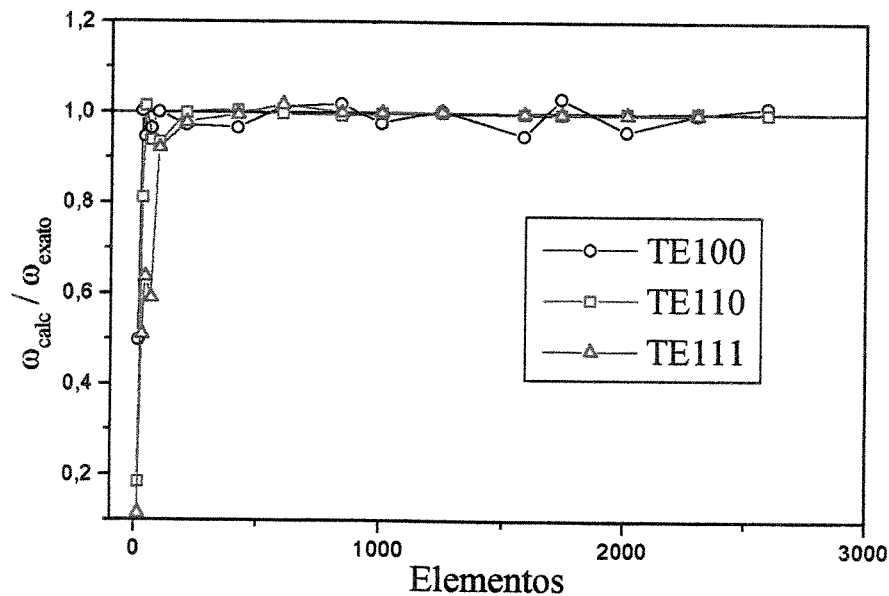


Figura 6.53 – Convergência dos resultados em relação ao resultado analítico, em números normalizados.

No gráfico da Figura 6.53, mostramos a frequência de ressonância normalizada (frequência calculada dividida pela frequência analítica) em função do número de elementos para três modos: TE_{100} , TE_{110} e TE_{111} . A linha reta foi adicionada apenas para facilitar a análise da convergência dos resultados.

Na Tabela 6.1, mostramos os valores teóricos e calculados para diversos modos TE. Ambos valores das frequências teórica e calculada são dados em radianos por segundo (rad/s) e devem ser multiplicados por 10^8 .

Modos TE_{lmn}	ω (teórico)	ω (calculado)	Erro (%)
100	9,4247	9,5533	1,36
010	23,5619	23,6365	0,31
001	31,4159	31,2859	0,41
110	25,3769	25,3974	0,08
101	32,7991	32,7765	0,06
011	39,2699	39,2537	0,04
111	40,3850	40,3742	0,02

Tabela 6.1 – Resultados teóricos e dos cálculos das frequências de ressonância para diversos modos TE, e a percentagem de erro entre elas.

Através deste gráfico podemos acompanhar o progresso dos resultados à medida que aumentamos o número de elementos da malha, e vemos que o programa de cálculo possui um comportamento oscilante que tende a diminuir à medida que aumentamos o número de incógnitas. Podemos notar que, em alguns casos, a convergência é bastante rápida, bastando poucos elementos para que o programa produza resultados satisfatórios. A convergência para os modos estudados foi atingida com um total de 2589 elementos e 622 nós, sendo que o erro máximo nos cálculos foi de 1,36%.

Com este exemplo, conseguimos ver que embora a qualidade da malha final não seja a máxima esperada, os resultados numéricos obtidos indicam uma baixa taxa de erro, não comprometendo a confiabilidade do resultado numérico.

Capítulo 7

Conclusões

Esta tese apresentou um novo algoritmo para o refinamento de malhas de elementos triangulares bidimensionais, e sua adaptação para estruturas tridimensionais superficiais e volumétricas.

7.1 – Técnicas de Refinamento

Esta tese apresentou primeiramente um algoritmo de refinamento bidimensional que combina duas técnicas de refinamento bastante conhecidas: Divisão baricêntrica e Divisão através da bissecção da maior aresta.

Como mostramos na Seção 2.3.3, esta nova técnica é superior às outras estudadas, pois mantém as vantagens delas, descartando suas desvantagens. Os resultados obtidos nos mostram que as malhas obtidas com este algoritmo são de boa qualidade geométrica e são, na pior hipótese, semelhantes ao melhor resultado obtido pelas outras técnicas estudadas. O

tempo de processamento também se aproxima bastante da técnica mais veloz estudada. O algoritmo também se mostrou capaz de produzir bons resultados quando tratamos de estruturas com grandes variações no tamanho dos seus elementos, mantendo as suas fronteiras internas. As simulações numéricas conduzidas utilizando malhas de elementos finitos produziram resultados extremamente satisfatórios, mostrando que as mesmas não possuem apenas boa qualidade geométrica, mas também computacional.

Também adaptamos esta nova técnica, com algumas restrições, ao refinamento tridimensional de superfícies, com resultados extremamente satisfatórios, tanto em relação à qualidade geométrica da malha, quanto em relação à qualidade dos resultados de simulações numéricas conduzidas com as malhas obtidas pelo nosso método. O tempo de processamento destas malhas também é bastante reduzido, próximo ao obtido no caso bidimensional.

A adaptação desta técnica ao refinamento de estruturas tridimensionais volumétricas infelizmente não produziu uma qualidade geométrica tão boa quanto a obtida nos casos anteriores. A implementação de técnicas de transformação topológica podem corrigir esta falha. Um dos pontos positivos desta implementação é a ausência de elementos tipo lasca, altamente indesejáveis. As simulações numéricas, feitas com malhas tetraédricas, mostram que elas conduzem a resultados com erros menores que 2%.

7.2 – Sugestões para Trabalhos Futuros

Um dos pontos aonde o algoritmo bidimensional falha ocorre quando temos elementos de aspecto ruim entre duas fronteiras internas, como mostrado na Figura 6.20. Técnicas para a correção destes elementos devem ser estudadas, para garantir uma malha totalmente funcional.

O uso da suavização das posições dos nós superficiais em estruturas tridimensionais não planas (esferas, etc...) também deve ser mais desenvolvido, para evitar uma possível destruição destas superfícies em refinamento.

A implementação das técnicas de transformação topológica em nosso algoritmo irá aumentar a qualidade geométrica dos mesmos; entretanto ainda é necessário um estudo mais profundo destas técnicas com o propósito de otimizá-las. Também é necessário um estudo detalhado da técnica de refinamento 3D volumétrica para poder otimizar o seu tempo de processamento, que ainda pode ser reduzido.

Outro fator interessante de se acrescentar ao nosso algoritmo, é o uso do processamento paralelo, para diminuir o tempo de processamento do mesmo.

Apêndice A

Estruturas de Dados e Arquivos de Entrada

Neste apêndice, iremos descrever a estrutura de dados utilizada para manusear as informações necessárias para o funcionamento dos programas de refinamento.

A.1 – Estruturas de Dados

Podemos dividir as estruturas de dados para refinamento em duas classes principais: As estruturas baseadas no armazenamento de arestas e as estruturas baseadas no armazenamento de triângulos. No nosso caso utilizamos o armazenamento baseado na descrição dos triângulos, pois conforme apontado em [Shewchuk96,97], esta representação acelera os processos de troca de diagonais, bissecção de elementos, entre outros. Esta aceleração é fundamentada no fato de que quando necessitamos fazer uma mudança estrutural em uma dada triangulação, a quantidade de tempo necessária para se ler e

escrever os dados de uma estrutura baseada em triângulos é menor que o tempo gasto para a mesma tarefa em uma estrutura de arestas. Além disso o acesso à conectividade de um triângulo é facilmente obtida através dos dados de um triângulo.

A.1.1 – Estruturas de Dados para o Refinamento Bidimensional

A estrutura utilizada para representar um determinado elemento consiste em um vetor composto de 7 ítems, que apesar de compacta é bastante funcional, abrangendo todos os dados necessários. Os 3 primeiros contêm os pontos que formam o triângulo. Os 3 itens seguintes se referem aos seus elementos vizinhos imediatos. O item final é utilizado para monitorar o refinamento nos casos em que ocorrem interfaces internas. Na figura A.1 vemos a representação de um elemento (E), os nós que o formam (p_1, p_2, p_3) e seus vizinhos imediatos (V_1, V_2, V_3).

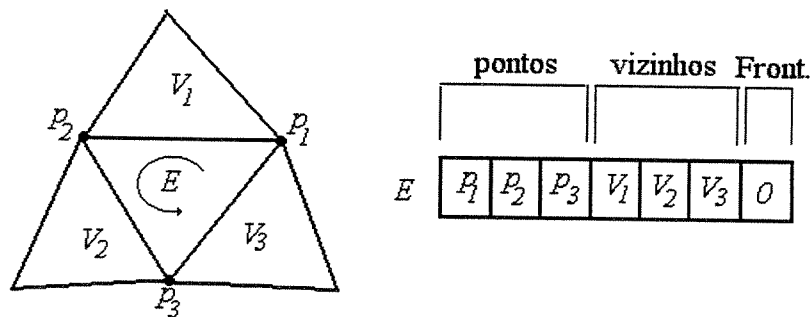


Figura A.1 – Representação do Elemento E em nossa estrutura de dados.

Apesar de bastante simples, esta representação possui alguns detalhes que devem ser respeitados, a fim assegurar a conformidade da malha e a correta execução do programa. Os pontos estão organizados de forma tal que a aresta formada por p_1p_2 é a maior aresta do triângulo, enquanto que os elementos vizinhos estão organizados de maneira que o elemento que compartilha a aresta p_1p_2 com o elemento E esteja na posição V_1 . Seguindo este raciocínio, o elemento que compartilha a aresta p_2p_3 estará na posição V_2 e o elemento que compartilha a aresta p_3p_1 estará na posição V_3 . Como podemos ver, os

dados estão organizados em uma direção anti-horária. Assim, se desejarmos colocar o ponto p_2 na primeira posição do vetor, devemos alterar a estrutura toda ficando então com a seguinte configuração: $(p_2 p_3 p_1)(V_2 V_3 V_1)$. Logicamente, estas convenções devem ser respeitadas também pelos elementos vizinhos. Por exemplo, o elemento V_1 na figura A.1 seria formada por: $(p_2 p_1 p_x)(E V_x V_y)$, aonde p_x , V_x , V_y são os pontos e elementos vizinhos correspondentes.

Outro detalhe da estrutura é o controle das fronteiras internas. Devido ao uso da troca de diagonais de Delaunay, é necessário definirmos quais fronteiras queremos preservar. Quando um determinado nó pertence a uma fronteira, adicionamos um sinal negativo ao mesmo nos vetores aos quais pertencem. Quando tivermos dois elementos com sinais negativos no vetor de um elemento, então a aresta correspondente não deve ser alterada, exceto para os casos de bissecção, quando, então, o novo nó também pertencerá à fronteira. Em um exemplo bastante simples, supondo que a aresta p_3p_1 , na figura A.1, está em uma fronteira, então a configuração do elemento E será: $(-p_1 p_2 -p_3)(V_1 V_2 V_3)$, enquanto que a configuração do elemento V_1 será dada por $(p_2 -p_1 p_x)(E V_x V_y)$. O último dado do vetor pode assumir um número inteiro que irá definir se esta fronteira interna pertence a uma reta, a um círculo, etc., sendo designado um número para cada caso. Logicamente, o programa deve estar capacitado para reconhecer as fronteiras curvas e os dados correspondentes devem ser fornecidos pelo usuário no arquivo de entrada. Na figura A.1, o 0 indica que se trata de uma fronteira reta.

É claro que devemos manter também dados a respeito das coordenadas dos nós, que são armazenadas em dois vetores diferentes, um para as coordenadas x e outro para as coordenadas y .

A.1.2 – Estruturas de Dados para o Refinamento 3D Superficial

No refinamento superficial de estruturas tridimensionais, utilizamos os mesmos dados básicos discutidos na seção anterior. Porém, agora, necessitamos acrescentar ao

menos mais dois dados para que possamos definir a qual superfície pertence este elemento e de que tipo é esta superfície. O primeiro dado é necessário uma vez que uma determinada estrutura pode ser formada por diversas superfícies, cada uma com características diferentes uma da outra, que são definidas pelo segundo marcador. A conexão entre as superfícies pode ser monitorada através da teoria de grafos, por exemplo [Souza93]. As interfaces entre uma superfície e outra podem ser definidas pelo marcador final do vetor apresentado na seção anterior. Este marcador também pode ser utilizado para definir fronteiras internas que desejamos preservar sobre uma superfície específica, utilizando o mesmo processo descrito na seção anterior.

Obviamente, é necessário acrescentar mais um vetor aos nossos vetores de posição x e y , para descrever as coordenadas z dos nós

A.1.3 – Estruturas de Dados para o Refinamento 3D Volumétrico

As estruturas de dados para o refinamento tridimensional volumétrico são um pouco diferentes das utilizadas nas seções anteriores. Como agora tratamos de malhas extremamente difíceis de se visualizar, e até mesmo de se imaginar, é natural que o grau de complexidade também aumente, e a estrutura de dados deve refletir estas mudanças.

Nas seções anteriores a estrutura de dados básica padrão definia um elemento por seus pontos e elementos vizinhos. Agora definiremos um elemento pelos seus pontos e pelas arestas que o compõe, e incluiu-se um vetor para cada aresta, que relaciona os pontos que a compõe e os elementos aos quais pertence. Esta mudança foi escolhida para agilizar o processo de refinamento e das transformações topológicas, como veremos mais adiante. O nosso vetor, que representa cada elemento, será então formado por 11 itens: os 4 itens iniciais são os nós que formam o tetraedro. Os 6 seguintes definem as arestas formadas pelos 4 pontos. Estas arestas são ordenadas em ordem decrescente de comprimento, a fim de facilitar a procura do elemento com a maior aresta. Como agora não é possível definirmos uma simetria no elemento, da forma como fazíamos nos conjuntos de dados das

seções anteriores, devido à própria natureza tridimensional do mesmo, não é necessária uma ordenação dos pontos. O último item refere-se a que volume pertence o elemento. É utilizado quando estamos fazendo o refinamento de estruturas com fronteiras internas.

As estruturas de dados que utilizam os elementos vizinhos a um dado elemento para definir a conectividade dos tetraedros, precisam de algoritmos de busca para definir quais elementos tem uma determinada aresta em comum. Com nossa estrutura, isto é feito automaticamente, bastando verificar quais os elementos aos quais ela pertence no vetor correspondente. Isto é muito valioso no processo de busca da maior aresta para o refinamento, pois uma vez determinada a maior aresta do elemento a ser dividido, basta consultar os elementos que possuem esta aresta em comum e verificar se eles possuem alguma aresta de diâmetro maior.

Na figura A.2 mostramos um exemplo simples desta classificação. Nela mostramos dois elementos, aonde o plano formado pelos nós $p_1p_2p_3p_4$ está em uma fronteira externa, que dividem entre si as arestas A_2 , A_4 e A_8 . Nesta figura mostramos, também, as representações do elemento E_1 e da aresta A_2 .

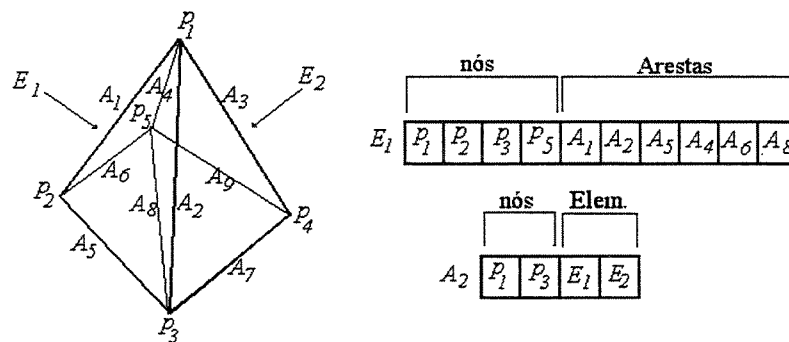


Figura A.2 – Representação do elemento E_1 e da Aresta A_2 .

Como notamos neste exemplo, este conjunto de dados também nos auxilia na hora de fazermos uma troca de diagonais na superfície de uma estrutura, pois basta procurarmos as arestas que tenham apenas dois elementos em sua estrutura. O mesmo princípio pode facilmente ser estendido para algumas transformações topológicas, como por exemplo, uma

aresta que pertença a 3 elementos é uma candidata à transformação $3 \rightarrow 2$, citada na seção 4.2, necessitando somente da verificação de que esta transformação é aplicável.

Como implementamos um processo de relaxação dos nós internos, devemos adicionar também um vetor que associe ao nó sua condição dentro da malha, isto é, que nos diga se este nó está em uma superfície, em uma junção entre duas superfícies ou em um vértice da estrutura, isto é, uma junção entre três ou mais superfícies, conforme seção 4.2. Isto é necessário para a prevenção de problemas no processo de relaxação da malha.

A.2 – Arquivos Iniciais

A seguir mostramos os arquivos iniciais necessários para cada tipo de refinamento: 2D, 3D superficial e 3D volumétrico.

Para o refinamento 2D são utilizados 3 arquivos:

- Um arquivo de controle que contém a maioria dos parâmetros da malha.
- Um arquivo que contém dados a respeito das gaussianas utilizadas para monitorar o refinamento.
- Um arquivo que contém os dados da malha inicial.

A seguir mostraremos um exemplo de cada tipo de arquivo e em seguida explicaremos cada item do mesmo.

Arquivo de Controle

```

** H M . I **
** inim ** iaut ** i2step(0/1)** n ** mx **
    1      1      1      1000  4900
** nt ** nmes ** inhm**
    0      1      0
*** archives: gaussians, elements data and points coordinates
gauss.dat      ,circle.dat
** nx **      ny **      delta **
    4      4      1.D-6
** xb (boundary coordinates: microns) **
-5.000 -1.000  1.000  5.000
** xd (divisions) **
    4      2      4
** yb (boundary coordinates: microns) **
-5.000 -1.000  1.000  5.000
** yd (divisions) **
    4      2      4
** Boxes
      1
-1.000 -1.000  1.000  1.000
#####

```

Elementos do arquivo de controle

Ini: Pode assumir 3 opções que definem como vai ser a nossa malha inicial.

- 1 – Malha automática. O próprio programa gera uma malha estruturada retangular, seguindo os dados fornecidos pelo usuário. Ver em dados para malha automática.
- 2 – Malha externa. É fornecida uma malha inicial pelo usuário.
- 4 – Malha pré-gravada. O programa lê uma malha já produzida pelo próprio programa. Ver *i2step*.

Iaut: Quando igual a 1 ativa o refinamento.

12step: Quando igual a 1 ativa o salvamento de uma malha para leitura posterior.

N : Número de elementos desejado.

Mx : Número máximo de elementos.

Nt: Ativa a medida do tempo de refinamento.

Nmes: Ativa as medidas de qualidade, ângulo interno, etc... da malha.

Inhm: Opções que definem como o refinamento será monitorado:

0 – Monitoração através de Gaussianas.

1 – Monitoração através de um arquivo contendo os valores de campo obtidos através de um cálculo anterior.

2 – Monitoração através de estimativas de erro.

Archives: lista os arquivos que contêm as definições das gaussianas, usado quando *inhm* = 0 e o arquivo que contêm a malha inicial externa, usado quando *ini* = 2. No arquivo acima vemos dois exemplos de arquivos: 'gauss.dat' e 'circle.dat'. Ver *Arquivo de Gaussianas* e *Arquivo da Malha Inicial*.

Dados para malha automática

Nx: Número de fronteiras no eixo *x*.

Ny: Número de fronteiras no eixo *y*.

Delta: Define um fator de truncamento.

Xb: Define as posições de cada uma das *nx* fronteiras.

Xd: Define o número de divisões que há entre 2 fronteiras consecutivas.

Yb: Define as posições de cada uma das *ny* fronteiras.

Yd: Define o número de divisões que há entre 2 fronteiras consecutivas.

Boxes: Define as fronteiras internas de uma malha automática. No nosso caso temos 2 fronteiras internas. As coordenadas das fronteiras são definidas da seguinte forma:

<i>Nr. da caixa</i>	<i>Coord. X inicial</i>	<i>Coord. Y inicial</i>	<i>Coord. X final</i>	<i>Coord. Y final</i>
---------------------	-------------------------	-------------------------	-----------------------	-----------------------

No arquivo exemplo acima, o malhador irá gerar uma malha automática quadrada de dimensão 10,0 x 10,0, com uma região quadrada de 2,0 x 2,0 em seu interior, e irá refiná-la, sendo monitorada pelas gaussianas do arquivo 'Gauss.dat', até atingir 1000 elementos. Após isto ele salvará a malha para que possa ser re-utilizada, e salvará dados a respeito da qualidade da malha.

Arquivo de Gaussianas

FUNÇÕES DE DENSIDADE (GAUSSIANAS)									
**	a1	**	x1	**	y1	**	d1x	**	d1y
	10.00		0.00		0.00		0.06		0.06
**	a2	**	x2	**	y2	**	d2x	**	d2y
	0.00		1.00		1.0		3.90		3.90
**	a3	**	x3	**	y3	**	d3x	**	d3y
	0.00				0.00		0.60		1.00
**	a4	**	x4	**	y4	**	d4x	**	d4y
	0.00				21.00		21.00		20.00
**	a5	**	x5	**	y5	**	d5x	**	d5y
	0.00				8.00		8.00		2.00
**	a6	**	x6	**	y6	**	d6x	**	d6y
	0.00				60.00		60.00		40.00
**	a7	**	x7	**	y7	**	d7x	**	d7y
	0.00				100.00		0.00		40.00
**	a8	**	x8	**	y8	**	d8x	**	d8y
	0.00				100.00		100.00		1.00
**	circle (core)				**				
**	a1	**	bb1	**	xco	**	aa1	**	dd1
	3.00				3.00		10.00		0.00
**	ellipse				**				
**	a2	**		**	b2	**	aa2	**	dd2
	2.00				3.00		0.00		0.05

Elementos do Arquivo de Gaussianas

O programa pode definir até 8 gaussianas diferentes e, como cada gaussiana possui o mesmo formato, definiremos apenas uma delas:

A1: Amplitude da gaussiana.

X1, Y1: Coordenadas do centro da gaussiana.

D1x, D1y: Distância do centro para a qual a amplitude da gaussiana cairá pela metade (Raio de decaimento).

Circle e Ellipse: Fornecem os dados de círculos e elipses, para os casos de fronteiras internas. Também monitoram o refinamento em torno destes círculos e elipses.

A1: Raio do círculo.

BB1: Raio auxiliar do círculo (não é mais utilizado).

Xco: Afastamento do centro do círculo no eixo x em relação ao centro (0.0, 0.0).

AA1: Amplitude das Gaussianas em torno do círculo.

DD1: Raio de decaimento.

A2: Define o tamanho do eixo da elipse em x.

B2: Define o tamanho do eixo da elipse em y .

AA2 e DD2: O mesmo que **AA1** e **DD1**.

Este arquivo exemplo irá gerar uma gaussiana de amplitude 10,0 centrada em (0,0,0), com raio de decaimento 0,6 microns. Se o nosso programa utilizar fronteiras circulares ou elípticas, as primeiras terão raio igual a 3,0 micrômetros e estarão a 10,0 micrômetros da origem, enquanto que a fronteira elíptica terá um raio em x de 2,0 μm e um raio em y de 3,0 μm .

Arquivo da Malha Inicial

```
**      E L T O P      **
** nel      **      np      ** list(1) **
**      3      5      1
**      eltop      **
**      1      -1      4      5      2      0      0      2
**      2      -1      -2      4      0      3      1      2
**      3      -2      3      4      0      0      2      2
**      x      y      **
**      1      0.0      3.0
**      2      2.0      0.0
**      3      10.0      0.0
**      4      10.0      10.0
**      5      0.0      10.0
```

Elementos do Arquivo da Malha Inicial

Nel: número de elementos da malha

Np: número de nós da malha

List: número de listagens. Será sempre igual a 1.

Eltop: Dados a respeito dos elementos. Segue a estrutura discutida no Apêndice A.1.1

X, Y: Cordenadas dos nós.

Os arquivos de entrada para o caso tridimensional superficial seguem o mesmo padrão, com a diferença de que incluímos a coordenada do centro e o raio de decaimento da gaussiana em relação a z . As definições para esfera permanecem as mesmas utilizadas para o círculo, onde introduzimos um novo termo (o raio em z) para definirmos um elipsóide, e

acrescentamos campos para a definição de cilindros e cones. No arquivo de controle, a geração automática da malha está desativada.

No caso do refinamento volumétrico, o arquivo de controle é do mesmo tipo mostrado, com a desativação da malha automática. As gaussianas são da mesma forma que no caso tridimensional superficial. A única mudança ocorre no arquivo da malha inicial, aonde na definição dos elementos, só precisamos definir os nós que compõem o elemento, e o material a que pertencem. Na definição das coordenadas dos nós é acrescentado um último item que indica a situação do nó na malha (0 = nó interno, 1 = nó superficial, 2 = nó de aresta e 3 = nó de canto).

Apêndice B

Interface Gráfica

O uso dos recursos computacionais no auxílio à visualização e montagem das malhas tem crescido muito nos últimos anos, principalmente devido à melhora da qualidade e capacidade de ‘software’ e ‘hardware’ [Preis95, Santo95, Vieira96, Souza96]. A Interface Gráfica desenvolvida durante esta tese consiste em um programa que auxilia na visualização, em tempo real, da malha a ser utilizada nas simulações através de elementos finitos. Como vimos anteriormente, é de fundamental importância para o cálculo dos campos eletromagnéticos que a malha inicial esteja bem construída, caso contrário o surgimento de incompatibilidades é quase certo. Como, em muitos casos, a geração da malha sem o auxílio visual do computador é um trabalho penoso e complicado, o objetivo desta interface é auxiliar na elaboração e visualização da malha, o que acarreta uma melhora na qualidade dos projetos.

O uso de técnicas específicas de programação e visualização permite que o programa atual, partindo de uma malha inicial, monitore as diversas variáveis que controlam o refinamento da malha, sem a necessidade de se sair do programa para alterar os arquivos de controle ou de gaussianas, como era feito até então [Rogers90, Foley97]. Assim, pode-se construir a malha desejada e visualizá-la em tempo real através de um só programa, o que não era possível antes.

A interface foi planejada para ser simples e prática, de forma a poder ser utilizada com o mínimo de treinamento possível. Portanto, optou-se por um sistema de janelas padrão para fazer a interface com o usuário, uma vez que este sistema já é amplamente conhecido [Newman95]. Utilizou-se, para a criação do programa, o software Digital Visual Fortran 5.0, que dispõe de recursos que possibilitam a criação de sistemas de janelas, gráficos, interatividade com o 'mouse', etc., de uma forma rápida, eficiente e segura, além de ser possível a utilização dos códigos já existentes em linguagem Fortran, sem a necessidade de atualizações e/ou tradução de uma linguagem para outra [Ribar93].

Na Figura B.1, vemos uma imagem inicial da nossa interface gráfica, na qual podemos ver seus diversos componentes e um exemplo de refinamento produzido com a ajuda desta interface.

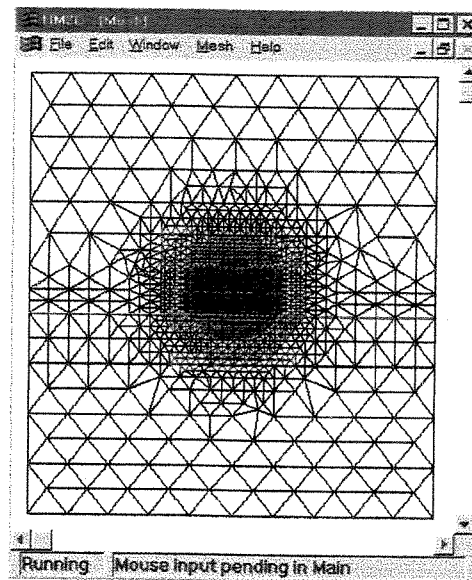


Figura B.1 – Interface gráfica e exemplo de malha.

Podemos ver que a interface gráfica consiste em uma barra de menus, uma barra de status e uma área de desenho que é utilizada para mostrar os dados que o programa repassa ao usuário durante o refinamento, e também para visualizar a malha obtida ao final do refinamento. Na barra de menus temos os menus: **File**, **Edit**, **Window**, **Mesh** e **Help**. A

seguir, damos um detalhamento maior a respeito de cada um dos menus e diálogos que compõem o programa.

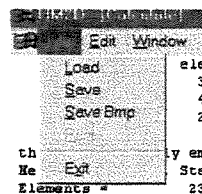


Figura B.2 – Menu *File* e seus componentes.

O menu *File*, que vemos na Figura B.2, permite acesso a arquivos de formato pré-definido, através do submenu *Load*, que aciona um diálogo para abertura de arquivos no formato padrão mostrado no Apêndice A, bem como a possibilidade de salvar os dados gerados, através do submenu *Save*, e também armazenar uma imagem em formato BMP da tela que estiver ativa no momento (*Save Bmp*). Também existem os recursos de impressão (*Print*) e a saída do programa, *Exit*.

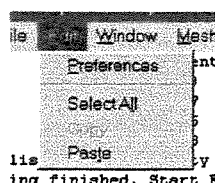


Figura B.3 – Menu *Edit* e seus componentes.

O menu *Edit*, que vemos na Figura B.3, permite a seleção, cópia e colagem da janela atual do programa, através dos submenus *Select All*, *Copy* e *Paste*. Este recurso pode ser usado quando se deseja exportar uma certa imagem da janela sem utilizar-se o recurso *Save Bmp*. Permite ainda a edição das preferências (*Preferences*) para utilização do nosso programa. Este menu abre uma caixa de diálogo com os diversos fatores usados para controlar a nossa malha, apresentados no Apêndice A. Esta caixa de diálogo é mostrada na Figura B.4.

Preferências

Malha Inicial: ☐ Automática ☒ Manual

Refinamento: **Refinamento**

Elementos: Nr. = 20 Máx. = 1000

Controle 2-Step: **Nenhum**

Medidas: **Nenhum**

Função de Densidade: **Gaussianas**

Arquivo - Gaussianas: gauss.dat **Edit**

Arquivo - Dados Eltop: circle.dat

Controles da Malha Automática

Interfaces - X	Coord. Interfaces - X	Divisões - X
4	-5.000 -1.000 1.000 5.000	4 2 4
Interfaces - Y	Coord. Interfaces - Y	Divisões - Y
4	0.000 5.000 6.200 10.000	4 1 3

Nr. of Boxes: 2

Coordenadas Box nr.: **-5.000 5.000 5.000 5.000**

OK **Cancel** **Save As...** **Help**

Figura B.4 – Diálogo de edição de preferências.

A nossa caixa de edição de preferências contém os seguintes dados que podem ser editados:

Malha Inicial – Automática: a malha inicial é gerada dentro do programa através de um algoritmo que utiliza os dados existentes em Controles da Malha Automática.

Manual: a malha inicial está em um arquivo, de formato pré-definido, que é lido pelo programa. O nome deste arquivo está na caixa Arquivo – Dados Eltop, que pode ser atualizada.

Refinamento - Caixa tipo Combo, no qual selecionamos se queremos fazer o Refinamento ou não.

Elementos – Neste módulo, podemos definir o número final desejado de elementos e a quantidade máxima de elementos que o programa irá gerar.

Controle 2-Step – Usado quando queremos ler e/ou escrever dados já calculados.

Medidas – Usado quando queremos fazer medidas de qualidade/ou tempo de cálculo.

Função de Densidade – Permite selecionarmos qual a função de densidade será usada nos cálculos: Gaussianas, que podem ser acessadas através do módulo Arquivo – Gaussianas,

Energia, proveniente de cálculos anteriores e Erro Estimado, que utiliza o resultado obtido com a estimativa de erro.

Arquivo – Gaussianas – Permite editarmos em qual arquivo estão as gaussianas que controlam o refinamento e modificar os valores destas gaussianas através do botão *Edit*, o que abre o diálogo mostrado na Figura B.5.

Arquivo – Dados Eltop – Permite editarmos em qual arquivo está a malha manual inicial.

Controles da Malha Automática – Contêm os elementos necessários para se gerar uma malha quadrada simples, que foram definidos no Apêndice A:

Interfaces X, Y – Número de interfaces que a malha terá.

Coordenadas das Interfaces X, Y – Coordenadas das interfaces em relação a seus eixos.

Divisões X, Y – Número de subdivisões entre duas interface.

Número de Boxes e Coordenadas – Permite que especifiquemos regiões dentro da malha que atuarão como interfaces que não podem ser alteradas pelo programa através da troca de diagonais, por exemplo.

Editar Gaussianas

	Amp.	X	Y	DX	DY
1-	0.00	0.00	0.80	0.06	0.07
2-	10.00	0.00	0.60	1.50	1.50
3-	0.00	0.00	0.60	1.00	1.00
4-	0.00	21.00	21.00	20.00	20.0
5-	0.00	8.00	8.00	2.00	2.00
6-	0.00	60.00	60.00	40.00	40.0
7-	0.00	100.00	0.00	40.00	40.0
8-	0.00	100.00	100.00	1.00	1.0

Circle

Rx	Ry	Xco	Amp.	D
3.00	3.00	10.00	0.00	0.50

Ellipse

Rx	Ry	Amp.	D
2.00	2.00	0.00	0.05

OK Cancel Help

Figura B.5 – Diálogo para edição das gaussianas.

Esta caixa ainda permite que salvemos estes dados com outros nomes (*Save As*), caso isto seja preciso, além de contar com uma pequena ajuda para o usuário inexperiente, através do botão *Help*.

Conforme dito anteriormente, é possível alterar o formato das Gaussianas de dentro do programa. Através deste diálogo, Figura B.5, podemos fazer isto alterando amplitude (*Amp.*), Posição central (X , Y) e raio de decaimento (DX , DY) nas funções gaussianas. Quando lidamos com interfaces internas curvas ou elípticas, a mudança é feita através dos raios (R_x , R_y), afastamento no eixo x (X_{co}), Amplitude (*Amp.*) e raio de decaimento (D).

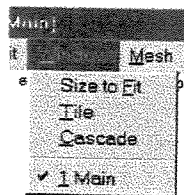


Figura B.6 – Menu *Window* e seus componentes.

O menu *Window*, visto na Figura B.6, permite alterações na apresentação das janelas fazendo com elas preencham totalmente a área da janela principal (*Size to Fit*), apareçam lado a lado (*Tile*) ou em cascata (*Cascade*). Possui ainda uma relação de todas janelas abertas até o momento.

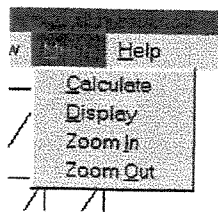


Figura B.7 – Menu *Mesh* e seus componentes.

O menu *Mesh*, visto na Figura B.7, permite iniciar o processo de refinamento (*Calculate*) e fazer a apresentação (*Display*) da malha. Possui ainda recursos de Zoom o

que é muito útil durante a visualização, pois permite vermos regiões pequenas da malha aonde podem estar ocorrendo concentrações de elementos. A seleção da região a ser aproximada é feita através do 'mouse', clicando-se em um ponto inicial, arrastando-se o mouse até a posição final e clicando-se novamente, o que define uma região retangular que será ampliada para preencher a área de trabalho.

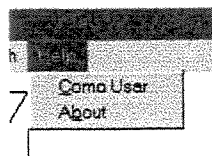


Figura B.8 – Menu *Help* e seus componentes.

Finalmente, através do menu *Help* obtemos informações sobre a barra de menus e suas funções e um pequeno diálogo contendo informações sobre versão do programa, etc.

Para a visualização de estruturas tridimensionais superficiais, a estrutura e apresentação do programa são basicamente as mesmas. A única mudança significativa é a introdução de um novo item no menu *Mesh* denominado *Rotate*, mostrado na Figura B.9, que nos possibilita acesso a um controle de rotação tridimensional da estrutura mostrada na tela.

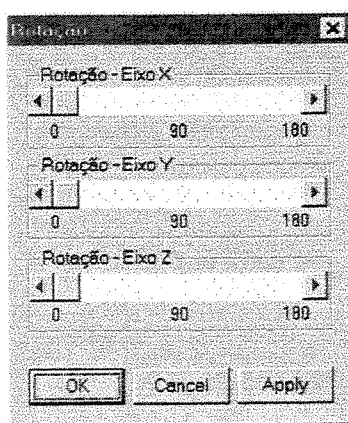


Figura B.9 – Diálogo de rotação da estrutura.

Para o programa de refinamento tridimensional volumétrico, incluímos ainda um novo item (*View Element*) que nos permite visualizar um determinado elemento, em vermelho na Figura B.10, e seus elementos vizinhos, em azul. Ele nos permite ainda rotacionar o elemento em torno do eixo z até acharmos a posição na qual o elemento é melhor apresentado.

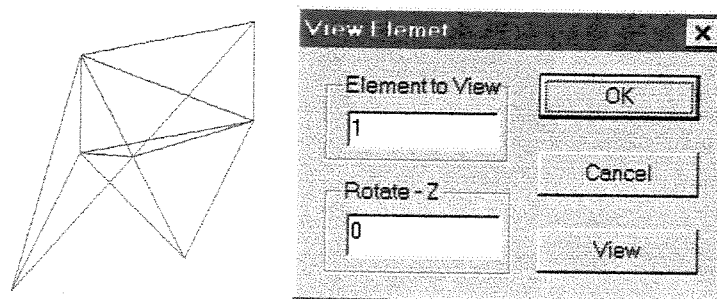


Figura B.10 – Diálogo para visualização de um determinado elemento, e um exemplo da apresentação do elemento.

Referências

Bajaj96 – C. L. Bajaj, E. J. Coyle & K. N. Lin. *Surface and 3D triangular meshes from planar cross section, Proc. of 5th International Meshing Roundtable, Sandia National Labs, pp.169-178.* (1996)

Bastos92 – J. P. A. Bastos. *Eletromagnetismo e Cálculo de Campos. Ed. Da UFSC (2^a Ed.).*(1992)

Baude96 – E. Baude. *Modelagem de um sensor baseado em uma fibra ótica curvada. Tese de Mestrado, CEFET-PR.* (1996)

Blacker96 – T. D. Blacker, R. J. Cass, S. E. Benzley & R. J. Meyers. *Generalized 3-D paving: an automatized quadrilateral surface mesh algorithm, International Journal for Numerical Methods in Engineering, vol.39, pp.1475-1489.* (1996)

Bomholt94 – L. H. Bomholt. *A 3D Finite Edge Element Code for Electromagnetics, from www.iis.ee.ethz.ch/research/1994/sim/bomho3/index.html.* (1994)

Borouchaki96 – H. Borouchaki, P. L. George & S. H. Lo. *Optimal Delaunay Point Insertion, International Journal for Numerical Methods in Engineering, v. 39, pp.3407-3437.* (1996)

Borouchaki97 – H. Borouchaki & P. L. George. *Aspects of 2-D Delaunay Mesh generation, International Journal for Numerical Methods in Engineering, v.40, pp.1957-1975.* (1997)

Canann93 – S. A. Cannan, M. B. Stephenson & T. Blacker. *Optismoothing: An optimization-driven approach to mesh smoothing, Finite Element in Analysis and Design, v.13. pp.185-190.* (1993)

Canann97 – S. A. Cannan, Y. C. Liu & A. V. Mobley. *Automatic 3D Surface meshing to address today's industrial needs, Finite Elements on Analysis and Design, vol.25, pp.185-198.* (1997)

Chen97 – H. Chen & J. Bishop. *Delaunay triangulation for curved surfaces, Proceedings of 6th International Meshing Roundtable, Sandia National Labs., pp.115-127* (1997)

Chew93 – L. P. Chew. *Guaranteed quality mesh generation for curved surfaces, Proceedings of 9th Annual Computational Geometry, CA, USA.* (1993)

Collin66 – R. E. Collin. *Foundations of Microwave Engineering, Mc Graw Hill.* (1966)

Dean96 – A. Dean. *Non-Structured Mesh for the Finite Integral Method of Solving Maxwell's Equations. PhD Thesis, University College London.* (1996)

Dieguez89 – J. P. P. Dieguez. *Modelagem geométrica para computação gráfica. Ed. Achiamé.* (1989)

Digital97 – Digital Equipment Corporation. *Digital Fortran – Language Reference Manual. Digital Equip. Corp.* (1997)

Fernandez93 – **F. A. Fernandez, Y. C. Yong & R. D. Ettinger.** *A Simple Mesh Generator For 2-D Finite Element Calculations, IEEE Transactions on Magnetics, Vol.29, pp. 1882-1885.* (1993.)

Figueroa96 – **H. E. H. Figueroa & M. Silveira.** *Efficient Finite Element Modeling of Optical Waveguides with Arbitrary Curved Interfaces, Proceedings of TELEMO'96 – VII Brazilian Symposium on Microwaves and Optoelectronics & XIV Brazilian Symposium on Telecommunications, Curitiba-PR, pp.394-399* (1996)

Figueroa97 – **H. E. H. Figueroa & E. Baude.** *Efficient Adaptive Refinement for Triangular Element Meshes, Proceedings of COMPUMAG'97 - XI Conference on the Computation of Electromagnetic Fields, Rio de Janeiro-RJ, pp.263-264.*(1997)

Foley97 – **J. D. Foley, A. van Dam, S. K. Feiner & J. F. Hughes.** *Computer Graphics: Principles and Practice, Addison-Wesley, 2nd.Ed. in C* (1997)

Freitag95 – **L. A. Freitag, M. T. Jones & P. E. Plassmann.** *An efficient parallel algorithm for mesh smoothing. Proc. of 4th. Int. Mesh. Roundtable, Albuquerque, New Mexico, USA, pp.47-58.* (1995)

Fuchs97 – **A. Fuchs.** *Almost Regular Delaunay-triangulations, International Journal for Numerical Methods in Engineering, v.40, pp.4595-4610.* (1997)

Fuenmayor96 – **F. J. Fuenmayor & J. L. Oliver.** *Criteria to achieve nearly optimal meshes in the h-adaptive finite element method, International Journal for Numerical Methods in Engineering, vol.39, pp.4039-4061* (1996)

George99 – **P. L. George.** *Tet meshing: construction, optimization and adaptation, Proceedings of 8th International Roundtable, California, USA, pp.133-141.* (1999)

Golias94 – N. A. Golias & T. D. Tsiboukis. *An approach to refining three-dimensional tetrahedral meshes based on Delaunay transformations*, *Int. J. Num. Meth. Eng.*, v.37, pp.793-812 (1994)

Golias94b – N. A. Golias, A. G. Papagiannakis & T. D. Tsiboukis. *Efficient mode analysis with edge elements and 3D adaptive refinement*, *IEEE Trans. on Microwave Theory and Techniques*, v.42, n.1, pp.99-107. (1994)

Gonzalez96 – M. L. y Gonzalez & R. C. Mesquita. *A survey of finite element mesh generation techniques in 3D*, *Proc. of CB MAG 96 – Congresso Brasileiro de Eletromagnetismo*, pp.128-131. (1996)

Harrington68 – R. F. Harrington. *Field Computation by Moment Methods*, *Macmillan*.(1968)

Hayata88 – K. Hayata, M. Nagai & M. Koshiba. *Finite-Element formalism for nonlinear slab-guided waves*, *IEEE Transactions on Microwave Theory and Techniques*, v.36, n.7, pp.1207-1215. (1988)

Hehl86 – M. E. Hehl. *Linguagem de programação estruturada Fortran 77*. *McGraw-Hill*.(1986)

Hinton91 – E. Hinton, N. V. R. Rao & M. Özakça. *Mesh generation with adaptive finite element analysis*, *Advanced Engineering Software*, vol.13, n.5/6, pp.238-262.(1991)

Hitschfeld95 – N. Hitschfeld. *Algorithms and data structures for handling a fully flexible refinement approach in mesh generation*. *Proceedings of 4th International Roundtable, Sandia National Laboratories*, pp.265-276. (1995)

Ho-Le88 – K. Ho-Le. *Finite Element Mesh Generation Methods: A Review and Classification*, *Computer-Aided Design*, v.20, n.1, pp.27-38 (1988)

Ida94 – N. **Ida** & K. C. **Chellamuthu**. '*A posteriori*' element by element local error estimation technique and 2D & 3D adaptive finite element mesh refinement. **IEEE Transactions on Magnetics**, vol.30, n.5, pp.3527-3530. (1994)

Itoh96 – T. **Itoh**, G. **Pelosi** & P. P. **Silvester** (eds.). *Finite Element Software for Microwave Engineering*. **John Wiley and Sons** (1996)

Jin93 – J. **Jin**. *The Finite Element Method in Eletromagnetics*. **John Wiley and Sons**. (1993)

Kettunen95 – L. **Kettunen** & K. **Forsman**. *Tetrahedral mesh generation in convex primitives*. **Int. J. Num. Meth. Eng.**, v.38, pp.99-117. (1995)

Koshiba92 – M. **Koshiba**. *Optical Waveguide Theory by the Finite Element Method*. **Kluwer Academic Publishers**. (1992)

Lee86 – D. L. **Lee**. *Eletromagnetic Principles of Integrated Optics*. **John Wiley & Sons**. (1986)

Lee95 – J. F. **Lee**. *Vector Finite Element Methods for Solving Electromagnetic Field Problems*. **PIERS'95**. (1995)

Lee99 – C. K. **Lee** & R. E. **Hobbis**. *Automatic Adaptive Finite Element Mesh Generation over Arbitrary two-dimensional Domain using Advancing Front Technique*, **Computers & Structures**, v. 71, pp.9-34 (1999)

Leithold82 – L. **Leithold**. *O Cálculo com geometria analítica*. **Ed. Harbra**. (1982)

Li92 – Q. Y. **Li**, R. A. **Sammur** & C. **Pask**. *Variational and finite element analyses of nonlinear strip opticals waveguides*. **Optics Communications**, 94, pp.37-43. (1992)

Mackerle93 – **J. Mackerle**. *Mesh generation and refinement for FEM and BEM – A bibliography (1990-1993)*. *Finite Element in Analysis and Design*, 15, pp.177-188. (1993)

Magalhães96 – **A. L. C. C. Magalhães & R. C. Mesquita**. *Solid Modeling application in Electromagnetism. Proc. of CB MAG 96 – Congresso Brasileiro de Eletromagnetismo*, pp.147-150. (1996)

Morcillo98 - **A. D. Morcillo, A. B. Ros & L. Nuño**. *Mesh generation over plane and curved surfaces. Proc. of 7th International Meshing Roundtable, Sandia National Labs, USA*, pp.397-407 (1998)

Nambiar93 - **R. V. Nambiar, R. S. Valera & K. L. Lawrence**. *An Algorithm For Adaptive Refinement Of Triangular Element Meshes*, *International Journal for Numerical Methods in Engineering*, vol.36, pp.499-509. (1993)

Newman95 – **W. M. Newman & M. G. Lamming**. *Interactive System Design*, Ed. Addison-Wesley. (1995)

Oden72 – **J. T. Oden**. *Finite Elements of Nonlinear Continua*, Ed. McGraw-Hill. (1972)

Peraire88 – **J. Peraire, J. Peiro, L. Formaggia, K. Morgan & O. C. Zienkiewicz**. *Finite element Euler computations in three dimensions. International Journal for Numerical Methods in Engineering*, vol.26, pp.2135-2159. (1988)

Preis95 – **K. Preis, I. Bardi, O. Biro, R. Hoschek, M. Mayr, U. Peterlini & K. R. Richter**. *Computer animation of eletromagnetic phenomena. IEEE Trans. on Magnetics*, v.31, n.3, pp.1714-1717. (1995)

Rahman91 – **B. M. A. Rahman, F. A. Fernandez & J. B. Davies.** *Review of Finite Element Methods for microwave and optical waveguides. **Proceedings of the IEEE**, v.79, n.10, pp.1442-1448.* (1991)

Redwine95 – **C. Redwine.** *Upgrading to Fortran 90. **Springer-Verlag*** (1995)

Renhart96 – **W. Renhart, W. Rucker & O. Biro.** *Calculation of antenna near field reactions on low conducting materials using finite element method, **IEEE Transactions on Magnetics**, v.31, n.3, pp.862-866* (1996)

Ribar93 – **L. J. Ribar.** *Fortran programming for Windows, Ed. **McGraw-Hill**.* (1993)

Rieder86 – **W. G. Rieder & H. R. Busby.** *Introductory Engineering Modeling Emphasizing Differential Models and Computer Simulations. **John Wiley & Sons**.* (1986)

Rivara89 – **M. C. Rivara.** *Selective Refinement/Derefinement Algorithms for Sequences of nested Triangulations, **International Journal for Numerical Methods in Engineering**, vol.28, pp.2889-2906.* (1989)

Rivara96 – **M. C. Rivara & M. Vemere.** *Cost Analysis of the Longest-Side (Triangle Bisection) Refinement Algorithm for Triangulation, **Engineering with Computers**, 12, pp.224-234.* (1996)

Rivara97a – **M. C. Rivara & P. Inostroza.** *Using Longest-Side Bisection Technique for the Automatic Refinement of Delaunay Triangulations. **International Journal for Numerical Methods in Engineering**, vol.40, pp.581-597.*(1997)

Rivara97b – **M. C. Rivara.** *New Longest-Edge Algorithms for the refinement and/or improvement of unstructured triangulations. **International Journal for Numerical Methods in Engineering**, vol.40, pp.3313-3324.* (1997)

Rogers90 – **D. F. Rogers & J. A. Adams.** *Mathematical elements for computer graphics.* Ed. McGraw-Hill, 2nd Ed.(1990)

Santo95 - **H. p. Santo (ed.).** *Graphics and Visualization for Finite Element Modeling, Special Issue of The International Journal of Applied Finite Elements and Computer Aided Design, Vol.19, N.º 4.* (1995).

Shephard88 – **M. S. Shephard.** *Approaches to the automatic generation and control of Finite element meshes, Applied Mechanical Review, v.41, pp.169-184.* (1988)

Shewchuk96 – **J. R. Shewchuk.** *Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. Proceed. Of 1st Workshop on Applied Computational Geometry, Philadelphia, Pennsylvania, EUA, pp.124-133.*(1996)

Shewchuk97 – **J. R. Shewchuk.** *Delaunay Refinement Mesh Generation, PhD Thesis, Carnegie Mellon University, Pittsburgh* (1997)

Silvester90 – **P. P. Silvester & R. L. Ferrari.** *Finite Element for Electrical Engineers, Cambridge University Press, 2nd Ed.* (1990)

Souza93 – **L. T. Souza & M. Gattass.** *A new scheme for mesh generation and mesh refinement using graph theory. Computer & Structures, vol.46, n.6, pp.1073-1084.*(1993)

Souza96 – **R. P. Souza & R. C. Mesquita.** *An advancing front mesh generator built inside Autocad. Proc. of CB MAG 96 – Congresso Brasileiro de Eletromagnetismo, pp.135-138.* (1996)

Spiegel73 – **M. R. Spiegel.** *Manual de fórmulas e tabelas matemáticas. Ed. McGraw-Hill.* (1973)

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Thalmann90 – **D. Thalmann (Ed.)**. *Scientific Visualization and graphics simulation*. **Ed. John Wiley & Sons**. (1990)

Thompson99 – **J. F. Thompson, B. K. Soni & N. P. Weatherill (Eds.)**. *Handbook of Grid Generation*, **CRC Press** (1999)

Tristano98 – **J. R. Tristano, S. J. Owen & S. A. Canann**. *Advancing front surface mesh generation in parametric space using a Riemannian surface*. **Proc. of 7th International Meshing Roundtable, Sandia National Labs**, pp.429-445. (1998)

Tsuji98 – **Y. Tsuji & M. Koshiba**. *Simple and efficient adaptive mesh generation for approximate scalar guided-mode and beam propagation solutions*. **IEICE Trans. Electron.**, v.E81-C, n.12, pp.1814-1820. (1998)

Venturini89 – **J. J. Venturini**. *Álgebra Vetorial e Geometria Analítica*. **Ed. Unificado**. (1989)

Vieira96 – **C. Vieira & J. M. Machado**. *An implementation of the advancing front algorithm for Windows environment*. **Proc. of CB MAG 96 – Congresso Brasileiro de Eletromagnetismo**, pp.132-134. (1996)

Wang91 – **J. J. H. Wang**. *Generalized Moment Methods in Electromagnetics*, **Ed. John Wiley & Sons**. (1991)

Yagel95 - **R. Yagel, D.S. Ebert, J. N. Scott & Y. Kurzion**. *Grouping Volume Renderers for Enhanced Visualization in Computational Fluid Dynamics*. **IEEE Transactions on Visualization and Computer Graphics**, Vol.1, No.2, pp. 117-131. (1995)

Zhu91 – **J. Z. Zhu & O. C. Zienkiewicz**. *Adaptivity and mesh generation*. **International Journal for Numerical Methods in Engineering**, vol.32, pp.783-810. 1991)